Commun. Comput. Phys.
doi: 10.4208/cicp.310513.170314a

Vol. **16**, No. 4, pp. 841-891
October 2014

# A Nominally Second-Order Cell-Centered Finite Volume Scheme for Simulating Three-Dimensional Anisotropic Diffusion Equations on Unstructured Grids

Pascal Jacq[1], Pierre-Henri Maire[1,*] and Rémi Abgrall[2]

[1] *CEA/CESTA, 15 Avenue des Sablières CS 60001, 33116 Le Barp cedex, France.*
[2] *Institüt für Mathematik, Universität Zürich, CH-8057 Zürich, Switzerland.*

Received 31 May 2013; Accepted (in revised version) 17 March 2014

Available online 24 July 2014

**Abstract.** We present a finite volume based cell-centered method for solving diffusion equations on three-dimensional unstructured grids with general tensor conduction. Our main motivation concerns the numerical simulation of the coupling between fluid flows and heat transfers. The corresponding numerical scheme is characterized by cell-centered unknowns and a local stencil. Namely, the scheme results in a global sparse diffusion matrix, which couples only the cell-centered unknowns. The space discretization relies on the partition of polyhedral cells into sub-cells and on the partition of cell faces into sub-faces. It is characterized by the introduction of sub-face normal fluxes and sub-face temperatures, which are auxiliary unknowns. A sub-cell-based variational formulation of the constitutive Fourier law allows to construct an explicit approximation of the sub-face normal heat fluxes in terms of the cell-centered temperature and the adjacent sub-face temperatures. The elimination of the sub-face temperatures with respect to the cell-centered temperatures is achieved locally at each node by solving a small and sparse linear system. This system is obtained by enforcing the continuity condition of the normal heat flux across each sub-cell interface impinging at the node under consideration. The parallel implementation of the numerical algorithm and its efficiency are described and analyzed. The accuracy and the robustness of the proposed finite volume method are assessed by means of various numerical test cases.

**AMS subject classifications**: 65M08, 65F10, 68W10, 76R50

**Key words**: Finite volume methods, unstructured grids, anisotropic diffusion, parallel computing.

*Corresponding author. *Email addresses:* `pascal.jacq@inria.fr` (P. Jacq), `maire@celia.u-bordeaux1.fr` (P.-H. Maire), `remi.abgrall@math.uzh.ch` (R. Abgrall)

# 1 Introduction

In this paper, we describe a finite volume scheme to solve anisotropic diffusion equations on unstructured grids. This three-dimensional scheme is the natural extension of the two-dimensional scheme CCLAD (Cell-Centered LAgrangian Diffusion) initially presented in [28]. Let us mention that it is also a 3D extension of the scheme proposed by Le Potier [34]. We aim at developing a robust and flexible method for diffusion operators devoted to the numerical modeling of the coupling between heat transfers and fluid flows. More precisely, we are concerned by the numerical simulation of heat transfers in the domain of hypersonic ablation of thermal protection systems [12]. In this context, one has to solve not only the compressible Navier-Stokes equations for the fluid flow but also the anisotropic heat equation for the solid materials which compose the thermal protection. These two models, *i.e.*, the Navier-Stokes equations and the heat equation, are strongly coupled by means of a surface ablation model which describes the removal of surface materials resulting from complex thermochemical reactions such as sublimation. We point out that in our case, the Navier-Stokes equations are solved employing a cell-centered finite volume method and the thermal protection system consists of several distinct materials with discontinuous conductivity tensors. This leads to the following requirements related to the diffusion scheme under consideration:

- It should be a finite volume scheme wherein the primary unknown, *i.e.*, the temperature is located at the cell center.
- It should be a sufficiently accurate and robust scheme to cope with unstructured three-dimensional grids composed of tetrahedral and/or hexahedral cells.

Before describing the main features of our finite volume scheme, let us briefly give an overview of the existing cell-centered diffusion scheme on three-dimensional grids. The simpler cell-centered finite volume is the so-called two-point flux approximation wherein the normal component of the heat flux at a cell interface is computed using the finite difference of the adjacent temperatures. It is well known that this method is consistent if and only if the computational grid is orthogonal with respect to the metrics induced by the symmetric positive definite conductivity tensor. This flaw renders this method inoperative for solving anisotropic diffusion problems on three-dimensional unstructured grids. It has motivated the work of Aavatsmark and his co-authors to develop a class of finite volume schemes based on multi-point flux approximations (MPFA) for solving the elliptic flow equation encountered in the context of reservoir simulation, refer to [2,3]. In this method, the flux is approximated by a multi-point expression based on transmissibility coefficients. These coefficients are computed using the pointwise continuity of the normal flux and the temperature across cell interfaces. The link between lowest-order mixed finite element and multi-point finite volume methods on simplicial meshes is investigated in [40]. The class of MPFA methods is characterized by cell-centered unknowns and a local stencil. The global diffusion matrix corresponding to this type of schemes

on general 3D unstructured grids is in general non-symmetric. There are many variants of the MPFA methods which differ in the choices of geometrical points and control volumes employed to derive the multi-point flux approximation. For more details about this method and its properties, the interested reader might refer to [4, 5, 18, 32] and the references therein. It is also worth mentioning that the theoretical analysis of the MPFA O scheme for heterogeneous anisotropic diffusion problems on general meshes have been performed in [7]. In this paper, the introduction of an hybrid discrete variational formulation and of a sufficient local condition for coercivity, depending on the grid and on the conductivity tensor, allows to prove the convergence of the proposed numerical method.

The mimetic finite difference (MFD) methodology is an interesting alternative approach for solving anisotropic diffusion equations on general unstructured grids. This method mimics the essential underlying properties of the original continuum differential operators such as conservation laws, solution symmetries and the fundamental identities of vector and tensor calculus, refer to [23–25,36,37]. More precisely, the discrete flux operator is the negative adjoint of the discrete divergence in an inner scalar product weighted by the inverse of the conductivity tensor. The classical MFD methods employ one degree of freedom per element to approximate the temperature and one degree of freedom per mesh face to approximate the normal component of the heat flux. The continuity of temperature and of the normal component of the heat flux across cell interfaces allows to assemble a global linear system satisfied by face-based temperatures unknowns. The corresponding matrix is symmetric positive definite. This type of discretization is usually second-order accurate for the temperature unknown on unstructured polyhedral grids having degenerate and non convex polyhedra with flat faces [26]. In the case of grids with strongly curved faces the introduction of more than one flux per curved face is required to get the optimal convergence rate [14].

Another class of finite volume schemes for solving diffusion equations, with full tensor coefficients, on general grids has been initially proposed in [19] and generalized in [20]. This approach has been termed discrete duality finite volume (DDVF) [16] since it arises from the construction of discrete analogs of the divergence and flux operators which fulfill the discrete counterpart of vector calculus identities. The DDFV method requires to solve the diffusion equation not only over the primal grid but also over a dual grid. Namely, there are both cell-centered and vertex-centered unknowns. In addition, the construction of the dual grid in the case of a three-dimensional geometry is not unique. There are at least three different choices which lead to different variants of the three-dimensional DDFV schemes, see [21] and the references therein. The DDFV method described in [21] is characterized by a symmetric definite positive matrix and exhibits a numerical second-order accuracy for the temperature. Compared to a classical cell-centered finite volume scheme, this DDFV discretization necessitates twice as much degrees of freedom over hexahedral grids [22]. Let us point out that the use of such a method might be difficult in the perspective of solving coupled problems such as heat transfer and fluid flow.

Finally, we note that besides the finite volume approach the finite element based formulations provide high-order reliable numerical methods for solving both time independent and time dependent diffusion problems, refer for instance to [31] and to [35] where this approach is applied for the equations of three-dimensional magnetic diffusion.

The main feature of our finite volume scheme relies on the partition of each polyhedral cell of the computational domain into sub-cells and on the partition of each cell face into sub-faces, which are composed of triangular faces. There is one degree of freedom per element to approximate the temperature unknown and one degree of freedom per sub-face to approximate the normal component of the heat flux across cell interfaces. For each cell, the sub-face normal fluxes impinging at a vertex are expressed with respect to the difference between sub-face temperatures and the cell-centered temperature. This approximation of the sub-face fluxes results from a local variational formulation written over each sub-cell. The sub-face temperatures, which are auxiliary unknowns, are locally eliminated by invoking the continuity of the temperature and the normal component of the heat flux across each cell interface. This elimination procedure of the sub-face temperatures in terms of the cell-centered temperatures surrounding a vertex is achieved by solving a linear system of reasonable size at each vertex. Gathering the contribution of each vertex allows to construct easily the global sparse diffusion matrix. The node-based construction of our scheme provides a natural treatment of the boundary conditions. The scheme stencil is local and for a given cell consists of the cell itself and its neighbors in the sense of nodes. Since the constitutive law of the heat flux has been approximated by means of a local variational formulation, the corresponding discrete diffusion operator inherits the positive definiteness property of the conductivity tensor. In addition, the semi-discrete version of the scheme is stable with respect to the discrete $L^2$ norm. For tetrahedral grids, the scheme preserves linear solutions with respect to the space variable and is characterized by a numerical second-order convergence rate for the temperature. For smooth distorted hexahedral grids its exhibits an accuracy which is almost of second-order. Let us point out that our formulation is similar to the local MFD discretization developed in [27] for simplicial grids.

The remainder of this paper is organized as follows. In Section 2, we first give the problem statement introducing the governing equations, the notation and assumptions for deriving our finite volume scheme. This is followed by Section 3, which is devoted to the space discretization of the scheme. In this section, we derive the sub-face fluxes approximation by means of a sub-cell-based variational formulation. We also describe the elimination of the sub-face temperatures in terms of the cell-centered unknowns to achieve the construction of the global discrete diffusion operator. The time discretization is briefly developed in Section 4. We describe the parallel implementation of the scheme and its efficiency in Section 5. Finally, the robustness and the accuracy of the scheme are assessed using various representative test cases in Section 6.

# 2 Problem statement

## 2.1 Governing equations

Our motivation is to describe a finite volume scheme that solves the anisotropic heat conduction equation on three-dimensional unstructured grids. This computational method is the natural extension to three-dimensions of the finite volume scheme that has been initially derived in [28]. Let us introduce the governing equations, notations and the assumptions required for the present work. Let $\mathcal{D}$ be an open set of the three-dimensional space $\mathfrak{R}^3$. Let $\boldsymbol{x}$ denotes the position vector of an arbitrary point inside the domain $\mathcal{D}$ and $t>0$ the time. We aim at constructing a numerical scheme to solve the following initial-boundary-value problem for the temperature $T=T(\boldsymbol{x},t)$

$$\rho C_v \frac{\partial T}{\partial t} + \nabla\cdot\boldsymbol{q} = \rho r, \qquad (\boldsymbol{x},t)\in\mathcal{D}\times[0,\mathcal{T}], \tag{2.1a}$$

$$T(\boldsymbol{x},0) = T^0(\boldsymbol{x}), \qquad \boldsymbol{x}\in\mathcal{D}, \tag{2.1b}$$

$$T(\boldsymbol{x},t) = T^*(\boldsymbol{x},t), \qquad \boldsymbol{x}\in\partial\mathcal{D}_D, \tag{2.1c}$$

$$\boldsymbol{q}(\boldsymbol{x},t)\cdot\boldsymbol{n} = q_N^*(\boldsymbol{x},t), \qquad \boldsymbol{x}\in\partial\mathcal{D}_N, \tag{2.1d}$$

$$\alpha T(\boldsymbol{x},t) + \beta\boldsymbol{q}(\boldsymbol{x},t)\cdot\boldsymbol{n} = q_R^*(\boldsymbol{x},t), \qquad \boldsymbol{x}\in\partial\mathcal{D}_R. \tag{2.1e}$$

Here, $\mathcal{T}>0$ denotes the final time, $\rho$ is a positive real valued function, which stands for the mass density of the material. The source term, $r$, corresponds to the specific heat supplied to the material and $C_v$ denotes the heat capacity at constant volume. We assume that $\rho$, $C_v$, and $r$ are known functions. The initial condition is characterized by the initial temperature field $T^0$. Three types of boundary conditions are considered: Dirichlet, Neumann and Robin boundary conditions. They consist in specifying respectively the temperature, the flux and a combination of them. We introduce the partition $\partial\mathcal{D}=\partial\mathcal{D}_D\cup\partial\mathcal{D}_N\cup\partial\mathcal{D}_R$ of the boundary domain. Here, $T^*$ and $q_N^*$ denote respectively the prescribed temperature and flux for the Dirichlet and Neumann boundary conditions. $\alpha$, $\beta$ and $q_R^*$ are the parameters of the Robin boundary condition. The vector $\boldsymbol{q}$ denotes the heat flux and $\boldsymbol{n}$ is the outward unit normal to the boundary of the domain $\mathcal{D}$.

Eq. (2.1a) is a partial differential parabolic equation of second order for the temperature $T$, wherein the conductive flux, $\boldsymbol{q}$, is defined according to the Fourier law

$$\boldsymbol{q} = -\mathbb{K}\nabla T, \tag{2.2}$$

where the second-order tensor, $\mathbb{K}$, is the conductivity tensor which is an intrinsic property of the material under consideration. We suppose that $\mathbb{K}$ is positive definite to ensure the model consistency with the Second Law of thermodynamics. Namely, this property ensures that heat flux direction is opposite to temperature gradient. Let us point out that in the problems we are considering the conductivity tensor is always symmetric positive definite, *i.e.*, $\mathbb{K}=\mathbb{K}^t$, where the superscript $t$ denotes transpose.

**Comment 2.1.** The normal component of the heat flux at the interface between two distinct materials, labeled by 1 and 2, is continuous, that is

$$(\mathbb{K}\nabla T)_1 \cdot \boldsymbol{n}_{12} = (\mathbb{K}\nabla T)_2 \cdot \boldsymbol{n}_{12},$$

where $\boldsymbol{n}_{12}$ is the unit normal to the interface. The temperature itself is also continuous.

## 2.2 Notations and assumptions

Let us introduce some notations that will be useful to develop the space discretization of problem (2.1). The domain $\mathcal{D}$ is paved with non overlapping polyhedral cells, *i.e.*, $\mathcal{D} = \cup_c \omega_c$, where $\omega_c$ denotes a generic polyhedral cell. In what follows, the letter $c$ will be used to denote quantities referring to the cell $\omega_c$. The list of vertices (points) of cell $c$ is denoted by $\mathcal{P}(c)$. Further, if $p$ is a generic point, its position vector is denoted by $\boldsymbol{x}_p$ and $\mathcal{C}(p)$ is the set of the cells surrounding it. In two-dimensional geometry the list of the counterclockwise ordered vertices belonging to a cell is sufficient to fully describe a cell. Unfortunately this is not the case anymore in three-dimensional geometry. To complete the cell geometry description, we introduce the set $\mathcal{F}(c)$ as being the list of faces of cell $c$ and the set $\mathcal{F}(p,c)$, which is the list of faces of cell $c$ impinging at point $p$. We observe that the former set is linked to the latter by $\mathcal{F}(c) = \cup_{p\in\mathcal{P}(c)} \mathcal{F}(p,c)$. A generic face is denoted either by the index $f$ or by $\partial\omega_c^f$.

Here, we consider a mesh composed of polyhedral cells. Namely, the term polyhedral cell stands for a volume enclosed by an arbitrary number of faces, each determined by an arbitrary number (3 or more) of vertices. If a face has four or more vertices, they can be non-coplanar, thus the face is not a plane and it is difficult to define its unit outward normal. To overcome this problem, we shall employ the decomposition of a polyhedral cell into elementary tetrahedra, initially introduced by Burton in [15] to discretize the conservation laws of Lagrangian hydrodynamics onto polyhedral grids. According to Burton's terminology, these elementary tetrahedra are called iotas, since $\iota$ is the smallest letter in the Greek alphabet. Being given the polyhedral cell $c$, we consider the vertex $p \in \mathcal{P}(c)$ which belongs to the face $f \in \mathcal{F}(c)$ and the edge $e$, refer to Fig. 1. The iota tetrahedron, $\mathcal{I}^{pfe}$, related to point $p$, face $f$ and edge $e$, is constructed by connecting point $p$, the centroid of cell $c$, the centroid of face $f$ and the midpoint of edge $e$ as displayed in Fig. 1. Further, we denote by $\boldsymbol{\mathcal{I}}^{pfec}$, the outward normal vector to the triangular face obtained by connecting the point $p$ to the midpoint of edge $e$ and the centroid of face $f$. Let us point out that $|\boldsymbol{\mathcal{I}}^{pfec}|$ is the area of the aforementioned triangular face.

Bearing this in mind, we can define the decomposition of the polyhedral cell, $\omega_c$, into sub-cells. The sub-cell, $\omega_{pc}$, related to point $p$ is obtained by gathering the iotas attached to point $p$ as follows

$$\omega_{pc} = \bigcup_{f\in\mathcal{F}(p,c)} \bigcup_{e\in\mathcal{E}(p,f)} \mathcal{I}^{pfe},$$

where $\mathcal{E}(p,f)$ is the set of edges of face $f$ impinging at point $p$. For the hexahedral cell displayed in Fig. 1, the sub-cell $\omega_{pc}$ is made of 6 iotas since there are 3 faces impinging

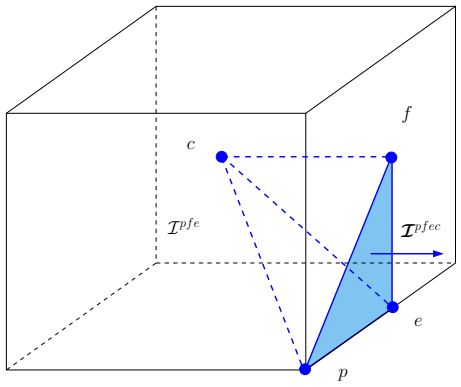



Figure 1: Definition of the iota cell $\mathcal{I}^{pfe}$ and the outward normal vector $\boldsymbol{\mathcal{I}}^{pfec}$ related to point $p$, face $f$ and edge $e$ in the hexahedral cell $c$.

at point $p$ and knowing that for each face there are two edges connected to point $p$. The volume of the sub-cell $\omega_{pc}$ is given by

$$|\omega_{pc}|=\sum_{f\in\mathcal{F}(p,c)}\sum_{e\in\mathcal{E}(p,f)}|\mathcal{I}^{pfe}|.$$

It is worth mentioning that the set of sub-cells, $\{\omega_{pc},\ p\in\mathcal{P}(c)\}$, is a partition of the polyhedral cell $c$ and thus the cell volume is defined by

$$|\omega_c|=\sum_{p\in\mathcal{P}(c)}|\omega_{pc}|.$$

The sub-face related to point $p$ and face $f$ is denoted by $\partial\omega_{pc}^f$ and defined as $\partial\omega_{pc}^f=\omega_{pc}\cap\partial\omega_c^f$. It consists of the union of the two outer triangular faces attached to the two iotas related to point $p$ and face $f$, refer to Fig. 2. The area and the unit outward normal corresponding to the sub-face $\partial\omega_{pc}^f$ are given by

$$A_{pc}^f=|\sum_{e\in\mathcal{E}(p,f)}\boldsymbol{\mathcal{I}}^{pfec}|,\quad \boldsymbol{n}_{pc}^f=\frac{1}{A_{pc}^f}\sum_{e\in\mathcal{E}(p,f)}\boldsymbol{\mathcal{I}}^{pfec}.$$

Let us point out that the set of sub-faces, $\{\partial\omega_{pc}^f,\ p\in\mathcal{P}(c,f)\}$, where, $\mathcal{P}(c,f)$ is the set of points of cell $c$ lying on face $f$, is a partition of the generic face $f$.

Now, we are in position to construct the space discretization of our diffusion problem. Integrating (2.1a) over cell $\omega_c$ and applying the divergence formula yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega_c}\rho C_v T(\boldsymbol{x},t)\mathrm{d}v+\int_{\partial\omega_c}\boldsymbol{q}\cdot\boldsymbol{n}\mathrm{d}s=\int_{\omega_c}\rho r(\boldsymbol{x},t)\mathrm{d}v. \tag{2.3}$$

Here, $\boldsymbol{n}$ is the unit outward normal to $\partial\omega_c$. The physical quantities $\rho, C_v$ and $r$ are supposed to be known functions with respect to space and time variables. We represent them

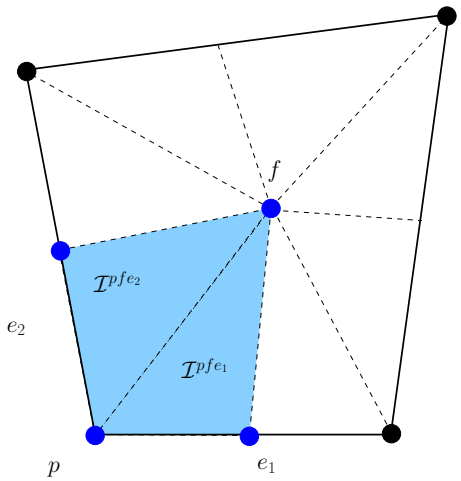



Figure 2: Generic quadrilateral face, $f$, related to the hexahedral cell $\omega_c$. The sub-face, $\partial\omega_{pc}^f$, related to point $p$ and face $f$ is obtained by gathering the triangular faces corresponding to the iotas $\mathcal{I}^{pfe_1}$ and $\mathcal{I}^{pfe_2}$.

with a piecewise constant approximation over each cell $\omega_c$. This approximation will be denoted with the subscript $c$. The conductivity tensor $\mathbb{K}$ is also considered to be constant over each cell and its approximation over $\omega_c$ is denoted by $\mathbb{K}_c$. Regarding the unknown temperature field, $T$, we are going to describe it using a piecewise constant approximation over each cell. Using these assumptions, (2.3) rewrites

$$m_c C_{vc} \frac{\mathrm{d}}{\mathrm{d}t} T_c + \int_{\partial\omega_c} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s = m_c r_c,$$

where $m_c$ is the mass of cell $\omega_c$, *i.e.*, $m_c = \rho_c|\omega_c|$, and $T_c = T_c(t)$ is the cell-averaged value of the temperature

$$T_c(t) = \frac{1}{|\omega_c|}\int_{\omega_c} T(\boldsymbol{x},t)\,\mathrm{d}v.$$

Finally, to achieve the first step of the space discretization of (2.3), it remains to discretize the surface integral of the heat flux employing the partition of faces into sub-faces. Knowing that $\partial\omega_c = \cup_{f\in\mathcal{F}(c)}\partial\omega_c^f$ the surface integral of the heat flux reads

$$\int_{\partial\omega_c} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s = \sum_{f\in\mathcal{F}(c)} \int_{\partial\omega_c^f} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s.$$

Now, recalling the partition of face $f$ into sub-cells, *i.e.*, $\partial\omega_c^f = \cup_{p\in\mathcal{P}(c,f)}\omega_{pc}^f$, leads to write the above surface integral as

$$\begin{aligned}\int_{\partial\omega_c} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s &= \sum_{f\in\mathcal{F}(c)}\ \sum_{p\in\mathcal{P}(c,f)} \int_{\omega_{pc}^f} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s \\ &= \sum_{p\in\mathcal{P}(c)}\ \sum_{f\in\mathcal{F}(p,c)} \int_{\omega_{pc}^f} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s.\end{aligned}$$

Here, we have interchanged the order of the double summation to finally get a global summation over the points of cell $c$ and a local summation over the faces impinging at point $p$. Let us denote by $q_{pc}^f$ the piecewise constant representation of the normal component of the heat flux over sub-face $\partial\omega_{pc}^f$

$$q_{pc}^f = \frac{1}{A_{pc}^f}\int_{\partial\omega_{pc}^f} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}s. \tag{2.4}$$

Gathering the above results, Eq. (2.3) turns into

$$m_c C_{vc}\frac{\mathrm{d}}{\mathrm{d}t}T_c + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{F}(p,c)} A_{pc}^f q_{pc}^f = m_c r_c. \tag{2.5}$$

To conclude this paragraph we introduce the sub-face temperature, which will be useful in the description of our scheme as auxiliary unknown

$$T_{pc}^f = \frac{1}{A_{pc}^f}\int_{\partial\omega_{pc}^f} T(\boldsymbol{x},t)\,\mathrm{d}s. \tag{2.6}$$

In writing this equation, we also assumed a piecewise constant approximation of the temperature field over each sub-face.

Let us write down the continuity conditions exposed at the end of the last section, in terms of sub-face fluxes and sub-face temperatures. To this end, we consider two neighboring cells denoted by $c$ and $d$ sharing a face and a point. The face is denoted by $f$ in the local list of faces of cell $c$ and $g$ in the local list of faces in cell $d$. Regarding the common point, it is denoted by $p$ in the local numbering of cell $c$ and $r$ in the local numbering of cell $d$. In what follows, we shall consider the sub-cells $\omega_{pc}$ and $\omega_{rd}$ sharing the sub-face $\partial\omega_{pc}^f \equiv \partial\omega_{rd}^g$, which is displayed in Fig. 3. For sake of simplicity, we have only plotted the common sub-face shared by the two sub-cells $\omega_{pc}$ and $\omega_{rd}$. When viewed from sub-cell $\omega_{pc}$ the sub-face temperature and the sub-face flux are denoted by $T_{pc}^f$ and $q_{pc}^f$, whereas viewed from sub-cell $\omega_{rd}$ they are denoted respectively by $T_{rd}^g$ and $q_{rd}^g$. Using the above notations and recalling that the unit outward normals satisfy $\boldsymbol{n}_{pc}^f = -\boldsymbol{n}_{rd}^g$ leads to write the continuity conditions for the temperatures and the heat flux as

$$A_{pc}^f q_{pc}^f + A_{rd}^g q_{rd}^g = 0, \tag{2.7a}$$

$$T_{pc}^f = T_{rd}^g. \tag{2.7b}$$

To achieve the space discretization of (2.5), it remains to construct an approximation of the sub-face normal flux, that is, to define a numeric sub-face flux function $h_{pc}^f$ such that:

$$q_{pc}^f = h_{pc}^f(T_{pc}^1 - T_c, \cdots, T_{pc}^k - T_c, \cdots, T_{pc}^{\mathsf{F}_{pc}} - T_c), \quad \forall f\in\mathcal{F}(p,c), \tag{2.8}$$

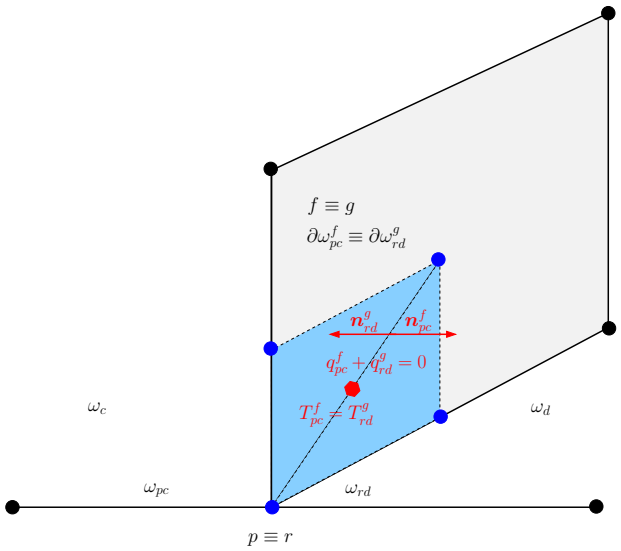



Figure 3: Continuity conditions for the sub-face fluxes and temperature on a sub-face shared by two sub-cells attached to the same point. Fragment of a polyhedral grid: quadrilateral face shared by hexahedral cells $c$ and $d$. Labels $p$ and $r$ denote the indices of the same point relatively to the local numbering of points in cell $c$ and $d$. The neighboring sub-cells are denoted by $\omega_{pc}$ and $\omega_{rd}$. They share the sub-face $\partial\omega_{pc}^f \equiv \partial\omega_{rd}^g$, which has been colored in blue.

where $\mathsf{F}_{pc}$ denotes the number of faces of cell $c$ impinging at point $p$, that is $\mathsf{F}_{pc} = |\mathcal{F}(p,c)|$.

To write our scheme we are going to define an approximation of the sub-face numerical fluxes in terms of sub-face temperatures and cell-centered temperatures. We shall then eliminate the sub-face temperatures using the continuity conditions (2.7) across the sub-faces interfaces. This is the topic of the next section.

# 3 Space discretization

We present the space discretization associated to our finite volume scheme, wherein the sub-face fluxes approximation results from a sub-cell-based variational formulation. Before proceeding any further, we start by giving a useful and classical result concerning the representation of a vector in terms of its normal components. This result leads to the expression of the standard inner product of two vectors, which will be one of the tools utilized to derive the sub-cell-based variational formulation.

## 3.1 Vector expression in terms of its normal components

Here, we describe the methodology to recover a three-dimensional vector at each vertex of a polyhedron from the normal components related to the sub-faces impinging at each vertex. Let $\boldsymbol{\phi}$ be an arbitrary vector of the three-dimensional space $\mathfrak{R}^3$ and $\boldsymbol{\phi}_{pc}$ its

piecewise constant approximation over the sub-cell $\omega_{pc}$. Let $\phi_{pc}^f$ be the sub-face normal components of $\boldsymbol{\phi}_{pc}$ defined by

$$\boldsymbol{\phi}_{pc}\cdot\boldsymbol{n}_{pc}^f=\phi_{pc}^f,\quad \forall f\in\mathcal{F}(p,c),$$

where $\mathcal{F}(p,c)$ is the set of sub-faces belonging to cell $c$ and impinging at point $p$. The above linear system is characterized by 3 unknowns, *i.e.*, the Cartesian components of the vector $\boldsymbol{\phi}_{pc}$ and $\mathsf{F}_{pc}=|\,\mathcal{F}(p,c)\,|$ equations. This system is properly defined provided that $\mathsf{F}_{pc}=3$. Namely, the number of faces of cell $c$, impinging at point $p$ must be strictly equal to 3. In what follows, we assume that the polyhedral cells we are working with are characterized by $\mathsf{F}_{pc}=3$. Let us remark that this restriction allows us to cope with tetrahedron, hexahedron and prism. The extension to the case $\mathsf{F}_{pc}>3$ will be investigated in Appendix B by studying the particular case of pyramids for which $\mathsf{F}_{pc}=4$ at one vertex.

Bearing this assumption in mind, let us introduce the corner matrix $\mathbb{J}_{pc}=[\boldsymbol{n}_{pc}^1,\boldsymbol{n}_{pc}^2,\boldsymbol{n}_{pc}^3]$ to rewrite the above $3\times3$ linear system as

$$\mathbb{J}_{pc}^t\boldsymbol{\phi}_{pc}=\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix},$$

where the superscript t denotes the transpose matrix. Granted that the vectors $\boldsymbol{n}_{pc}^f$, for $f=1,\cdots,3$, are not co-linear, the above linear system has always a unique solution, which reads

$$\boldsymbol{\phi}_{pc}=\mathbb{J}_{pc}^{-t}\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix}. \tag{3.1}$$

This equation allows to express any vector in terms of its normal components on the local basis $\{\boldsymbol{n}_{pc}^1,\boldsymbol{n}_{pc}^2,\boldsymbol{n}_{pc}^3\}$. This representation provides the computation of the inner product of two vectors $\boldsymbol{\phi}_{pc}$ and $\boldsymbol{\psi}_{pc}$ as follows

$$\boldsymbol{\phi}_{pc}\cdot\boldsymbol{\psi}_{pc}=\left(\mathbb{J}_{pc}^t\mathbb{J}_{pc}\right)^{-1}\begin{pmatrix}\psi_{pc}^1\\ \psi_{pc}^2\\ \psi_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix}. \tag{3.2}$$

A straightforward computation shows that the $3\times3$ matrix $\mathbb{H}_{pc}=\mathbb{J}_{pc}^t\mathbb{J}_{pc}$ is expressed in terms of the dot products of the basis vectors

$$\left(\mathbb{H}_{pc}\right)_{ij}=\boldsymbol{n}_{pc}^j\cdot\boldsymbol{n}_{pc}^i.$$

This matrix is symmetric positive definite and represents the local metric tensor associated to the sub-cell $\omega_{pc}$.

**Comment 3.1.** Let us remark that the problem of finding the expression of a vector in terms of its normal components always admits a unique solution in the two-dimensional case since the number of faces of cell $c$ impinging at point $p$ is always equal to two.

### 3.2 Sub-cell-based variational formulation

We construct an approximation of the sub-face fluxes by means of a local variational formulation written over the sub-cell $\omega_{pc}$. Contrary to the classical cell-based variational formulation used in the context of Mimetic Finite Difference Method [23,26,29], the present sub-cell-based variational formulation leads to a local **explicit expression of the sub-face fluxes** in terms of the sub-face temperatures and the mean cell temperature. The local and explicit feature of the sub-face fluxes expression is of great importance, since it allows to construct a numerical scheme with only one unknown per cell. We also want to mention that this method is the three-dimensional extension of the procedure initially developed in [28].

Our starting point to derive the sub-cell based variational formulation consists in writing the partial differential equation satisfied by the heat flux. From the heat flux definition (2.2), it follows that $\boldsymbol{q}$ satisfies

$$\mathbb{K}^{-1}\boldsymbol{q}+\nabla T=\mathbf{0}. \tag{3.3}$$

Let us point out that the present approach is strongly linked to the mixed formulation utilized in the context of mixed finite element discretization [6,27,38]. Dot-multiplying the above equation by an arbitrary vector $\boldsymbol{\phi}\in\mathcal{D}$ and integrating over the cell $\omega_{pc}$ yields

$$\int_{\omega_{pc}}\boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv=-\int_{\omega_{pc}}\boldsymbol{\phi}\cdot\nabla T dv,\quad\forall\boldsymbol{\phi}\in\mathcal{D}. \tag{3.4}$$

Integrating by part the right-hand side and applying the divergence formula lead to the following variational formulation

$$\int_{\omega_{pc}}\boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv=\int_{\omega_{pc}}T\nabla\cdot\boldsymbol{\phi}dv-\int_{\partial\omega_{pc}}T\boldsymbol{\phi}\cdot\boldsymbol{n}ds,\quad\forall\boldsymbol{\phi}\in\mathcal{D}. \tag{3.5}$$

This sub-cell-based variational formulation is the cornerstone to construct a local and explicit expression of the sub-face fluxes. Replacing $T$ by its piecewise constant approximation, $T_c$, in the first integral of the right-hand side and applying the divergence formula to the remaining volume integral leads to

$$\int_{\omega_{pc}}\boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv=T_c\int_{\partial\omega_{pc}}\boldsymbol{\phi}\cdot\boldsymbol{n}ds-\int_{\partial\omega_{pc}}T\boldsymbol{\phi}\cdot\boldsymbol{n}ds,\quad\forall\boldsymbol{\phi}\in\mathcal{D}. \tag{3.6}$$

Observing that the sub-cell boundary, $\partial\omega_{pc}$, decomposes into the inner part $\underline{\partial\omega_{pc}}=\partial\omega_{pc}\cap\omega_c$ and the outer part $\overline{\partial\omega_{pc}}=\partial\omega_{pc}\cap\partial\omega_c$ allows to split the surface integrals of the right-hand side of (3.6) as follows

$$\int_{\omega_{pc}}\boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv=T_c\int_{\overline{\partial\omega_{pc}}}\boldsymbol{\phi}\cdot\boldsymbol{n}ds+T_c\int_{\underline{\partial\omega_{pc}}}\boldsymbol{\phi}\cdot\boldsymbol{n}ds-\int_{\overline{\partial\omega_{pc}}}T\boldsymbol{\phi}\cdot\boldsymbol{n}ds-\int_{\underline{\partial\omega_{pc}}}T\boldsymbol{\phi}\cdot\boldsymbol{n}ds. \tag{3.7}$$

We replace $T$ by its piecewise constant approximation, $T_c$, in the fourth surface integral of the right-hand side, then noticing that the second integral is equal to the fourth one transforms Eq. (3.7) into

$$\int_{\omega_{pc}} \boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv = T_c\int_{\overline{\partial\omega_{pc}}}\boldsymbol{\phi}\cdot\boldsymbol{n}ds - \int_{\overline{\partial\omega_{pc}}} T\boldsymbol{\phi}\cdot\boldsymbol{n}ds. \tag{3.8}$$

**Comment 3.2.** A this point it is interesting to remark that the above sub-cell-based formulation is a sufficient condition to recover the classical cell-based variational formulation. This is due to the fact that the set of sub-cells is a partition of the cell, *i.e.*,

$$\omega_c = \bigcup_{p\in\mathcal{P}(c)} \omega_{pc}, \quad \text{and} \quad \partial\omega_c = \bigcup_{p\in\mathcal{P}(c)} \overline{\partial\omega_{pc}}.$$

Thus, summing (3.8) over all the sub-cells of $\omega_c$ leads to

$$\int_{\omega_c} \boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv = T_c\int_{\partial\omega_c}\boldsymbol{\phi}\cdot\boldsymbol{n}ds - \int_{\partial\omega_c} T\boldsymbol{\phi}\cdot\boldsymbol{n}ds.$$

This last equation corresponds to the cell-based variational formulation of the partial differential equation (3.3). This form is used in the context of Mimetic Finite Difference Method [23] to obtain a discretization of the heat flux. More precisely, it leads to a linear system satisfied by the sub-face fluxes. This results in a non explicit expression of the sub-face flux with respect to the sub-face temperatures and the cell-centered temperature [26], which leads to a finite volume discretization characterized by face-based and cell-based unknowns. In contrast to this approach, the sub-cell-based variational formulation yields a finite volume discretization with one unknown per cell.

We pursue the study of the sub-cell-based variational formulation discretizing the right-hand side of (3.8). First, we recall that the outer boundary of sub-cell $\omega_{pc}$ decomposes into sub-faces as

$$\overline{\partial\omega_{pc}} = \bigcup_{f\in\mathcal{F}(p,c)} \partial\omega_{pc}^f,$$

where $\partial\omega_{pc}^f$ is the sub-face associated to point $p$ and face $f$ in cell $c$, and $\mathcal{F}(p,c)$ is the set of faces of cell $c$ impinging at point $p$. Utilizing the above partition allows to rewrite the right-hand side of (3.8) as

$$\int_{\omega_{pc}} \boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv = T_c\sum_{f\in\mathcal{F}(p,c)}\int_{\partial\omega_{pc}^f}\boldsymbol{\phi}\cdot\boldsymbol{n}ds - \sum_{f\in\mathcal{F}(p,c)}\int_{\partial\omega_{pc}^f} T\boldsymbol{\phi}\cdot\boldsymbol{n}ds. \tag{3.9}$$

Introducing the sub-face temperature, $T_{pc}^f$, given by (2.6) and the sub-face approximation of vector $\boldsymbol{\phi}$ defined by $\phi_{pc}^f = \frac{1}{A_{pc}^f}\int_{\partial\omega_{pc}^f}\boldsymbol{\phi}\cdot\boldsymbol{n}\mathrm{d}s$, where $A_{pc}^f$ is the area of the sub-face, leads to rewrite the above equation as follows

$$\int_{\omega_{pc}} \boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv = -\sum_{f\in\mathcal{F}(p,c)} A_{pc}^f(T_{pc}^f - T_c)\phi_{pc}^f. \tag{3.10}$$

Finally, assuming a piecewise constant representation of the test function $\boldsymbol{\phi}$ allows to compute the volume integral in the left-hand side thanks to the quadrature rule

$$\int_{\omega_{pc}} \boldsymbol{\phi}\cdot\mathbb{K}^{-1}\boldsymbol{q}dv = w_{pc}\boldsymbol{\phi}_{pc}\cdot\mathbb{K}_c^{-1}\boldsymbol{q}_{pc}. \tag{3.11}$$

Here, $\mathbb{K}_c$ is the piecewise constant approximation of the conductivity tensor, $\boldsymbol{\phi}_{pc}$ and $\boldsymbol{q}_{pc}$ are the piecewise constant approximation of the vectors $\boldsymbol{\phi}$ and $\boldsymbol{q}$. In addition, $w_{pc}$ denotes some positive corner volume related to sub-cell $\omega_{pc}$, which will be determined later.

**Comment 3.3.** Let us note that the quadrature weight, $w_{pc}$, must satisfy the consistency condition

$$\sum_{p\in\mathcal{P}(c)} w_{pc} = |\omega_c|. \tag{3.12}$$

Namely, the corner volumes of a cell sum to the volume of the cell. This requirement ensures that constant functions are exactly integrated using the above quadrature rule.

Expressing the vectors $\boldsymbol{q}_{pc}$ and $\boldsymbol{\phi}_{pc}$ in terms of their normal components by means of (3.1) allows to write the right-hand side of (3.11) as

$$w_{pc}\boldsymbol{\phi}_{pc}\cdot\mathbb{K}_c^{-1}\boldsymbol{q}_{pc} = w_{pc}\left(\mathbb{J}_{pc}^t\mathbb{K}_c\mathbb{J}_{pc}\right)^{-1}\begin{pmatrix}q_{pc}^1\\ q_{pc}^2\\ q_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix}, \tag{3.13}$$

where $\mathbb{J}_{pc}$ is the corner matrix defined by $\mathbb{J}_{pc} = [\boldsymbol{n}_{pc}^1,\boldsymbol{n}_{pc}^2,\boldsymbol{n}_{pc}^3]$. Recalling that $|\mathcal{F}(p,c)| = 3$ leads to rewrite the right-hand side of (3.10)

$$-\sum_{f\in\mathcal{F}(p,c)} A_{pc}^f(T_{pc}^f - T_c)\phi_{pc}^f = -\begin{pmatrix}A_{pc}^1(T_{pc}^1-T_c)\\ A_{pc}^2(T_{pc}^2-T_c)\\ A_{pc}^3(T_{pc}^3-T_c)\end{pmatrix}\cdot\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix}. \tag{3.14}$$

Finally, combining (3.13) and (3.14), the sub-cell variational formulation becomes

$$w_{pc}\left(\mathbb{J}_{pc}^t\mathbb{K}_c\mathbb{J}_{pc}\right)^{-1}\begin{pmatrix}q_{pc}^1\\ q_{pc}^2\\ q_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix} = -\begin{pmatrix}A_{pc}^1(T_{pc}^1-T_c)\\ A_{pc}^2(T_{pc}^2-T_c)\\ A_{pc}^3(T_{pc}^3-T_c)\end{pmatrix}\cdot\begin{pmatrix}\phi_{pc}^1\\ \phi_{pc}^2\\ \phi_{pc}^3\end{pmatrix}. \tag{3.15}$$

Knowing that this equation must hold for any vector $\boldsymbol{\phi}_{pc}$, we obtain

$$\begin{pmatrix}q_{pc}^1\\ q_{pc}^2\\ q_{pc}^3\end{pmatrix} = -\frac{1}{w_{pc}}(\mathbb{J}_{pc}^t\mathbb{K}_c\mathbb{J}_{pc})\begin{pmatrix}A_{pc}^1(T_{pc}^1-T_c)\\ A_{pc}^2(T_{pc}^2-T_c)\\ A_{pc}^3(T_{pc}^3-T_c)\end{pmatrix}. \tag{3.16}$$

This equation constitutes the approximation of the sub-face normal fluxes. This local approximation is compatible with the expression of the constitutive law (2.2) in the sense

that the discrete approximation of the heat flux is equal to a tensor times the approximation of the temperature gradient. This tensor can be viewed as an effective conductivity tensor associated to the sub-cell $\omega_{pc}$. Thus, it is natural to set

$$\mathbb{K}_{pc} = \mathbb{J}_{pc}^t \mathbb{K}_c \mathbb{J}_{pc}.$$

Let us emphasize that this corner tensor inherits all the properties of the conductivity tensor $\mathbb{K}_c$. Namely, $\mathbb{K}_c$ being symmetric positive definite, $\mathbb{K}_{pc}$ is also symmetric positive definite. Recalling that $\mathbb{J}_{pc} = [\boldsymbol{n}_{pc}^1, \boldsymbol{n}_{pc}^2, \boldsymbol{n}_{pc}^3]$, we readily obtain the expression of the entries of the corner tensor, $\mathbb{K}_{pc}$, in terms of the unit normals $\boldsymbol{n}_{pc}^f$ for $f = 1, \cdots, 3$ and the cell conductivity $\mathbb{K}_c$

$$\left(\boldsymbol{K_{pc}}\right)_{fg} = (\boldsymbol{K_c} \boldsymbol{n}_{pc}^f) \cdot \boldsymbol{n}_{pc}^g.$$

Finally, the sub-face flux approximation for the sub-face $f$ is written under the compact form

$$q_{pc}^f = -\alpha_{pc} \sum_{g=1}^{3} \left(\mathbb{K}_{pc}\right)_{fg} A_{pc}^g (T_{pc}^g - T_c), \tag{3.17}$$

where $\alpha_{pc} = \frac{1}{w_{pc}}$.

**Comment 3.4.** We have followed exactly the construction of the sub-cell variational formulation described in [28]. The notations are a bit different to take into account the three-dimensional specificities, but the conclusions we have drawn so far are the same.

### 3.3 Inequality satisfied by the discrete sub-face normal flux approximation

In this paragraph we demonstrate that the discrete approximation of the sub-face normal fluxes (3.17) satisfies a discrete version of the fundamental inequality which follows from the Second Law of thermodynamics: $\boldsymbol{q} \cdot \nabla T \leq 0$. The discrete counterpart of the fundamental inequality states that for the sub-faces fluxes defined according to (3.17) the following inequality holds

$$\sum_{c \in \mathcal{C}(p)} \left( \sum_{f \in \mathcal{F}(p,c)} A_{pc}^f q_{pc}^f \right) T_c \geq 0. \tag{3.18}$$

To demonstrate this result, let us introduce, $I_p$, the nodal quantity defined by

$$I_p = \sum_{c \in \mathcal{C}(p)} \left( \sum_{f \in \mathcal{F}(p,c)} A_{pc}^f q_{pc}^f \right) T_c. \tag{3.19}$$

We prove that $I_p$ is always positive using the sub-cell variational formulation. Imposing $\boldsymbol{\phi} = \boldsymbol{q}$ in (3.15) yields

$$w_{pc} \mathbb{K}_{pc}^{-1} \begin{pmatrix} q_{pc}^1 \\ q_{pc}^2 \\ q_{pc}^3 \end{pmatrix} \cdot \begin{pmatrix} q_{pc}^1 \\ q_{pc}^2 \\ q_{pc}^3 \end{pmatrix} = - \begin{pmatrix} A_{pc}^1 (T_{pc}^1 - T_c) \\ A_{pc}^2 (T_{pc}^2 - T_c) \\ A_{pc}^3 (T_{pc}^3 - T_c) \end{pmatrix} \cdot \begin{pmatrix} q_{pc}^1 \\ q_{pc}^2 \\ q_{pc}^3 \end{pmatrix}. \tag{3.20}$$

Now, rearranging the right-hand side leads to

$$w_{pc}\mathbb{K}_{pc}^{-1}\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}=\left(\sum_{i=1}^{3}A_{pc}^i q_{pc}^i\right)T_c-\sum_{i=1}^{3}A_{pc}^i q_{pc}^i T_{pc}^i. \tag{3.21}$$

We notice that the left hand-side of (3.21) is always non-negative since $\mathbb{K}_{pc}$ is positive definite. Summing Eq. (3.21) over all cells surrounding $p$ yields

$$\sum_{c\in\mathcal{C}(p)} w_{pc}\mathbb{K}_{pc}^{-1}\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}=\sum_{c\in\mathcal{C}(p)}\left(\sum_{i=1}^{3}A_{pc}^i q_{pc}^i\right)T_c-\sum_{c\in\mathcal{C}(p)}\left(\sum_{i=1}^{3}A_{pc}^i q_{pc}^i T_{pc}^i\right). \tag{3.22}$$

Due to the continuity condition of the sub-face temperatures, the second term of the right-hand side is equal to zero. Finally, Eq. (3.22) becomes

$$I_p=\sum_{c\in\mathcal{C}(p)} w_{pc}\mathbb{K}_{pc}^{-1}\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}\cdot\begin{pmatrix}q_{pc}^1\\q_{pc}^2\\q_{pc}^3\end{pmatrix}\geq 0, \tag{3.23}$$

which ends the proof.

**Comment 3.5.** Inequality (3.23) is not only the discrete counterpart of the Second Law of thermodynamics but also the cornerstone to demonstrate the $L^2$-stability of the semi-discrete formulation of our finite volume scheme as we shall see in Section A.2.

### 3.4 Computation of the corner volume $w_{pc}$

We show that the sub-face normal fluxes approximation given by (3.16) preserves linear temperature fields over tetrahedral cells provided that the corner volume $w_{pc}$ is defined by $w_{pc}=\frac{1}{4}\,|\,\omega_c\,|$. To demonstrate this result, let us consider a generic tetrahedron, $\omega_c$, over which the temperature field, $T=T(\boldsymbol{x})$, is linear with respect to the space variable $\boldsymbol{x}$. The vertices of this tetrahedron are denoted respectively by $p$, $r$, $r+1$ and $r+2$, refer to Fig. 4. The temperatures at these vertices are $T_p$, $T_r$, $T_{r+1}$ and $T_{r+2}$. They coincide with the pointwise values of the linear temperature field. The constant value of the conductivity tensor over $\omega_c$ is $\mathbb{K}_c$. The heat flux is the constant vector $\boldsymbol{q}_c=-\mathbb{K}_c\nabla T$, which satisfies the identity

$$\boldsymbol{q}_c=-\frac{1}{|\,\omega_c\,|}\int_{\omega_c}\mathbb{K}_c\nabla T\,\mathrm{d}v.$$

Utilizing the divergence formula in the above equation turns it into

$$\boldsymbol{q}_c=-\frac{1}{|\,\omega_c\,|}\int_{\partial\omega_c}\mathbb{K}_c T\boldsymbol{n}\,\mathrm{d}s.$$

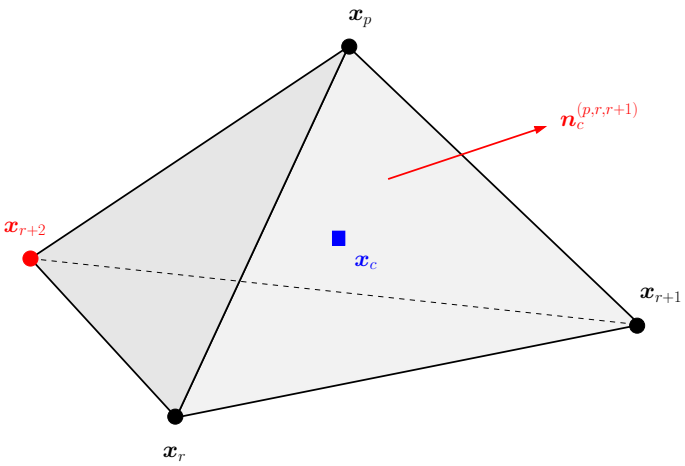



Figure 4: Generic tetrahedron with vertices $(x_p,x_r,x_{r+1},x_{r+4})$ and centroid $x_c=\frac{1}{4}(x_p+x_r+x_{r+1}+x_{r+2})$.

Now, expanding the surface integral over the triangular faces of the tetrahedral cell yields

$$q_c=-\frac{1}{|\omega_c|}\sum_{f\in\mathcal{F}(c)}\mathbb{K}_c A_c^f n_c^f \widetilde{T}_c^f,$$

where $A_c^f$ is the area of face $f$, $n_c^f$ is the unit outward normal to face $f$ and $\widetilde{T}_c^f$ is the face-averaged value of the temperature. This face-averaged temperature is computed by means of

$$\widetilde{T}_c^f=\frac{1}{3}\sum_{s\in\mathcal{P}(c,f)}T_s, \tag{3.24}$$

where $\mathcal{P}(c,f)$ is the set of points of cell $c$ belonging to face $f$. Before proceeding any further, we explicit our notations to highlight the role played by point $p$. Each triangular face is characterized by the set of its three vertices. The three faces impinging at point $p$ are $(p,r+k,r+k+1)$ for $k=1,\cdots,3$ and assuming a cyclic indexing. Their area, unit outward normal and face-averaged temperature are denoted respectively by $A_c^k$, $n_c^k$ and $\widetilde{T}_c^k$. The remaining face, which is opposite to point $p$, is $(r,r+1,r+2)$. Its area, unit outward normal and face-averaged temperature are denoted respectively by $A_c^r$, $n_c^r$ and $\widetilde{T}_c^r$. With the above notations, the heat flux expression becomes

$$q_c=-\frac{1}{|\omega_c|}\left(\sum_{k=1}^{3}\mathbb{K}_c A_c^k n_c^k \widetilde{T}_c^k+\mathbb{K}_c A_c^r n_c^r \widetilde{T}_c^r\right).$$

Knowing that $A_c^r n_c^r=-\sum_{k=1}^{3}A_c^k n_c^k$ leads to rewrite the above flux expression as

$$q_c=-\frac{1}{|\omega_c|}\sum_{k=1}^{3}\mathbb{K}_c A_c^k n_c^k\left(\widetilde{T}_c^k-\widetilde{T}_c^r\right).$$

Substituting the expression of the face-averaged temperatures (3.24) in terms of the point temperatures yields

$$\boldsymbol{q}_c = -\frac{1}{3|\omega_c|}\sum_{k=1}^{3}\mathbb{K}_c A_c^k \boldsymbol{n}_c^k \left(T_p - T_{r+k+2}\right).$$

Finally, to eliminate the point temperatures in the above expression, we introduce the cell-averaged temperature

$$T_c = \frac{1}{4}(T_p + T_{r+k} + T_{r+k+1} + T_{r+k+2}).$$

Due to the cyclic numbering, this expression is valid for $k = 1,\cdots,3$. Expressing $T_{r+k+2}$ in terms of the cell-averaged temperature and the remaining point temperatures leads to write

$$T_p - T_{r+k+2} = 4\left(\bar{T}_c^{(p,r+k,r+k+1)} - T_c\right),$$

where $\bar{T}_c^{(p,r+k,r+k+1)}$ is the sub-face temperature given by

$$\bar{T}_c^{(p,r+k,r+k+1)} = \frac{1}{4}(2T_p + T_{r+k} + T_{r+k+1}).$$

Since the temperature field is linear with respect to the space variable, we point out that the above expression is the exact value of the temperature field taken at the point $\bar{\boldsymbol{x}}_c^{(p,r+k,r+k+1)}$ located on the triangular face $(p,r+k,r+k+1)$, refer to Fig. 5, and defined by

$$\bar{\boldsymbol{x}}_c^{(p,r+k,r+k+1)} = \frac{1}{4}(2\boldsymbol{x}_p + \boldsymbol{x}_{r+k} + \boldsymbol{x}_{r+k+1}).$$

Observing the triangular face displayed in Fig. 5, we note that this point is the midpoint of the median segment coming from vertex $p$.

Gathering the above results allows to rewrite the expression of the heat flux as

$$\boldsymbol{q}_c = -\frac{4}{3|\omega_c|}\sum_{k=1}^{3}\mathbb{K}_c A_c^k \boldsymbol{n}_c^k \left(\bar{T}_c^{(p,r+k,r+k+1)} - T_c\right).$$

It remains to simplify the above expression of the heat flux by employing notations related to the sub-face associated to point $p$ and face $k$, displayed in blue color in Fig. 5. It is clear that the area of the sub-face, $A_{pc}^k$, is equal to one-third of the face area, $A_c^k$, and thus $A_c^k = 3A_{pc}^k$. In addition, the unit outward normal to the sub-face, $\boldsymbol{n}_{pc}^k$, coincides with the unit outward normal to the face, $\boldsymbol{n}_c^k$. Finally, defining the sub-face temperature $T_{pc}^k \equiv \bar{T}_c^{(p,r+k,r+k+1)}$ leads to write the heat flux

$$\boldsymbol{q}_c = -\frac{4}{|\omega_c|}\sum_{k=1}^{3}\mathbb{K}_c A_{pc}^k \boldsymbol{n}_{pc}^k \left(T_{pc}^k - T_c\right).$$

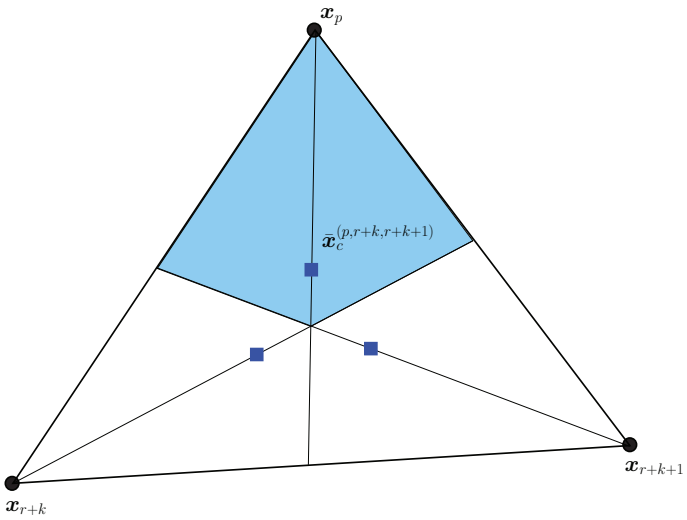



Figure 5: Triangular face $(p,r+k,r+k+1)$ related to the tetrahedron displayed in Fig. 4. The sub-face related to point $p$ has been colored in blue. The three degrees of freedom related to the sub-face temperatures are plotted by means of blue squares.

We dot-multiply the heat flux by the unit normal $\boldsymbol{n}_{pc}^l$ to obtain the normal component of the heat flux related to the sub-face $l$

$$q_{pc}^l = -\frac{4}{|\omega_c|}\sum_{k=1}^{3}\left(\mathbb{K}_c\boldsymbol{n}_{pc}^k\right)\cdot\boldsymbol{n}_{pc}^l A_{pc}^k\left(T_{pc}^k - T_c\right).$$

This formula coincides with the one derived from the variational formulation, refer to Eq. (3.17), provided that the volume weight satisfies $w_{pc} = \frac{1}{4}|\omega_c|$, which ends the proof.

This shows that the flux approximation (3.17) is exact for linear temperature fields with respect to the space variable. In addition, the sub-face temperatures coincide with the pointwise values taken by the linear temperature field at the midpoint of the median segment coming from each vertex of a triangular face. It is worth pointing out that this results has been already obtained in [27] using a more theoretical framework.

Finally, for general polyhedral cells, the corner volume weight related to sub-cell $\omega_{pc}$ is defined by

$$w_{pc} = \frac{1}{\mathsf{P}_c}|\omega_c|, \tag{3.25}$$

where $\mathsf{P}_c = |\mathcal{P}(c)|$ is the number of vertices of cell $\omega_c$.

## 3.5 Elimination of the sub-face temperatures

Having defined the flux approximation in terms of the difference between the cell and the sub-face temperatures, we shall express the sub-face temperatures in terms of the cell temperatures of the cells $c$ surrounding a specific point $p$, using the continuity conditions

of the normal heat flux at cell interfaces. In order to have a simpler expression of the equations we are going to introduce some new local notations. First of all, in this paragraph we are dealing with quantities located around a point $p$, so in all the notations we will omit to specify the subscript $p$. For each face $f$ in the list $\mathcal{F}(p)$ of the faces impinging at the node $p$ we associate two tuples $(c,i)$ and $(d,j)$ which identify the neighboring cells $c$ and $d$ of the face $f$ and their local numbering $i$ (resp. $j$) in the subset $\mathcal{F}(p,c)$ (resp. $\mathcal{F}(p,d)$) of $\mathcal{F}(p)$. With this notation a sub-face temperature $T^i_{pc}$ is denoted by $\bar{T}^i_c$ and using the continuity condition on the temperature is equal to $T^j_{pd}$ which is denoted $\bar{T}^j_d$ and can also be simply denoted by $\bar{T}^f$. The bar notation help us to make the difference between the cell centered unknown and the sub-face unknown. Similarly the area of the sub-face f can be indifferently noted $A^i_c$, $A^j_d$ or $A^f$. The local conductivity tensor $\mathbb{K}_{pc}$ will now be denoted by $\mathbb{K}^c$ so its components $\left(\mathbb{K}_{pc}\right)_{ij}$ can be written $\mathbb{K}^c_{ij}$.

Using this notation Eq. (3.17), which defines the heat flux approximation, rewrites

$$q^i_c = -\alpha_c \sum_{k=1}^{3} \mathbb{K}^c_{ik} A^k_c (\bar{T}^k_c - T_c), \tag{3.26}$$

where $\alpha_c$ is the inverse of the volume weight. The continuity condition of the sub-face fluxes across the face $f \equiv (c,i) \equiv (d,j)$ reads

$$A^i_c q^i_c + A^j_d q^j_d = 0.$$

Replacing the sub-face fluxes by their approximation (3.26) into the above equation yields

$$-\alpha_c A^i_c \sum_{k=1}^{3} \mathbb{K}^c_{ik} A^k_c (\bar{T}^k_c - T_c) - \alpha_d A^j_d \sum_{k=1}^{3} \mathbb{K}^d_{jk} A^k_d (\bar{T}^k_d - T_d) = 0.$$

Let us point out that this equation holds for all the faces $f$ impinging at node $p$, *i.e.* for all $f \in \mathcal{F}(p)$. Denoting $\mathsf{F}_p = |\mathcal{F}(p)|$ the number of faces impinging at node $p$, the set of all the above equations forms a $\mathsf{F}_p \times \mathsf{F}_p$ linear system, which writes under the compact form

$$\mathbb{N}\bar{\boldsymbol{T}} = \mathbb{S}\boldsymbol{T}. \tag{3.27}$$

Here, the matrix $\mathbb{N}$ is a $\mathsf{F}_p \times \mathsf{F}_p$ square matrix and $\bar{\boldsymbol{T}} \in \mathfrak{R}^{\mathsf{F}_p}$ is the vector of sub-face temperatures. Denoting $\mathsf{C}_p = |\mathcal{C}(p)|$ the number of cells surrounding node $p$, the matrix $\mathbb{S}$ is a $\mathsf{F}_p \times \mathsf{C}_p$ rectangular matrix and vector $\boldsymbol{T} \in \mathfrak{R}^{\mathsf{C}_p}$ is the vector of cell temperatures. The matrix $\mathbb{N}$ has five non-zero terms on each lines, its diagonal part writes

$$\mathbb{N}_{ff} = \alpha_c A^i_c \mathbb{K}^c_{ii} A^i_c + \alpha_d A^j_d \mathbb{K}^d_{jj} A^j_d.$$

Regarding its extra-diagonal parts, two terms comes from the contribution of the sub-cell $\omega_{pc}$. Let $g$ be a generic face of cell $c$ impinging at point $p$ characterized by the index $k$

in the local numbering, *i.e.*, $g \equiv (c,k)$, then the extra-diagonal entries related to cell $c$ and faces $i$ and $k$ write

$$\mathbb{N}_{fg} = \alpha_c A_c^i \mathbb{K}_{ik}^c A_c^k, \quad \text{for } k \in [1,3] \quad \text{and} \quad k \neq i.$$

The two remaining terms come from the sub-cell $\omega_{pd}$. Let $g$ be a generic face of cell $d$ impinging at point $p$ characterized by the index $k$ in the local numbering, *i.e.*, $g \equiv (d,k)$, then the extra-diagonal entries related to cell $d$ and faces $j$ and $k$ write

$$\mathbb{N}_{fg} = \alpha_d A_d^j \mathbb{K}_{jk}^d A_d^k, \quad \text{for } k \in [1,3] \quad \text{and} \quad k \neq j.$$

Let us remark that the matrix $\mathbb{N}$ has a symmetric structure, for $g \equiv (c,k)$, $f \equiv (c,i)$ we have $\mathbb{N}_{gf} = \alpha_c A_c^k \mathbb{K}_{ki}^c A_c^i$ and for $g \equiv (d,k)$, $f \equiv (d,j)$ we have $\mathbb{N}_{gf} = \alpha_d A_d^k \mathbb{K}_{kj}^d A_d^j$. We also note that $\mathbb{N}$ is symmetric if and only if $\mathbb{K}^c$ (resp. $\mathbb{K}^d$) is symmetric.

Finally, the matrix $\mathbb{S}$ has two non-zero terms on each row, one term for each neighboring cell $c$ and $d$ of the face $f$

$$\mathbb{S}_{fc} = \alpha_c \sum_{k=1}^{3} A_c^i \mathbb{K}_{ik}^c A_c^k,$$

$$\mathbb{S}_{fd} = \alpha_d \sum_{k=1}^{3} A_d^j \mathbb{K}_{jk}^d A_d^k.$$

It remains to investigate the properties of the matrices $\mathbb{N}$ and $\mathbb{S}$, this is the topic of the next section.

### 3.6 Properties of the matrices $\mathbb{N}$ and $\mathbb{S}$

The main motivation of this paragraph is to demonstrate the invertibility of the matrix $\mathbb{N}$ to ensure that the linear system (3.27) that solves the sub-face temperatures in terms of the cell temperatures admits always a unique solution. To this end, let us show that $\mathbb{N}$ is a positive-definite matrix. First, we introduce the matrix $\mathbb{L}^c$ of size $3 \times \mathsf{F}_p$ defined by

$$\mathbb{L}_{ij}^c = \begin{cases} 1, & \text{if } j \equiv (c,i), \\ 0, & \text{elsewhere.} \end{cases}$$

Here, $\mathbb{L}^c$ is the rectangular matrix which associates the sub-face of cell $c$ in its local numbering to its numbering around point $p$. Let us define the diagonal matrix $\mathbb{A}$ of size $\mathsf{F}_p \times \mathsf{F}_p$, which contains the area of the sub-faces, namely $\mathbb{A}_{ff} = A_i^c$ for the face $f \equiv (c,i)$. Let us define $\mathbb{A}^c = \mathbb{L}^c \mathbb{A}$, the matrix which relates the area of sub-face of cell $c$ in its local numbering to its numbering around the point $p$. Employing this notation, it is straightforward to show that matrix $\mathbb{N}$ writes

$$\mathbb{N} = \sum_{c \in \mathcal{C}(p)} \alpha_c (\mathbb{A}^c)^t \mathbb{K}^c \mathbb{A}^c.$$

We are going to show that $\mathbb{N}\bar{\boldsymbol{T}}\cdot\bar{\boldsymbol{T}}>0$, for all $\bar{\boldsymbol{T}}\in\mathfrak{R}^{\mathsf{F}_p}$. To this end, let us compute $\mathbb{N}\bar{\boldsymbol{T}}\cdot\bar{\boldsymbol{T}}$ employing the above decomposition of $\mathbb{N}$

$$\begin{aligned}\mathbb{N}\bar{\boldsymbol{T}}\cdot\bar{\boldsymbol{T}}&=\sum_{c\in\mathcal{C}(p)}\alpha_c(\mathbb{A}^c)^t\mathbb{K}^c\mathbb{A}^c\bar{\boldsymbol{T}}\cdot\bar{\boldsymbol{T}}\\&=\sum_{c\in\mathcal{C}(p)}\alpha_c\mathbb{K}^c(\mathbb{A}^c\bar{\boldsymbol{T}})\cdot(\mathbb{A}^c\bar{\boldsymbol{T}}).\end{aligned}$$

Recalling that $\alpha_c$ is non-negative and $\mathbb{K}^c$ is positive definite ensures that the right-hand side of the above equation is always non-negative, which ends the proof. Thus, matrix $\mathbb{N}$ is invertible and the sub-face temperatures are expressed in terms of the cell temperatures by means of the relation

$$\bar{\boldsymbol{T}}=\left(\mathbb{N}^{-1}\mathbb{S}\right)\boldsymbol{T}. \tag{3.28}$$

Further, if the cell temperature field is uniform, then the sub-face temperatures are also uniform and share the same constant value. This property follows from the relation satisfies by the matrices $\mathbb{N}$ and $\mathbb{S}$

$$\left(\mathbb{N}^{-1}\mathbb{S}\right)\mathbf{1}_{\mathsf{C}_p}=\mathbf{1}_{\mathsf{F}_p}, \tag{3.29}$$

Here, $\mathbf{1}_n$, where $n$ is an integer, is the vector of size $n$, whose entries are equal to 1. To demonstrate the above relation, let us show that $\mathbb{S}\mathbf{1}_{\mathsf{C}_p}=\mathbb{N}\mathbf{1}_{\mathsf{F}_p}$ by developing respectively the left and the right-hand side of this equality. Substituting the non-zero entries of matrix $\mathbb{S}$ leads to write the left-hand side

$$\begin{aligned}\left(\mathbb{S}\mathbf{1}_{\mathsf{C}_p}\right)_f&=\mathbb{S}_{fc}+\mathbb{S}_{fd}\\&=\alpha_c\sum_{k=1}^{3}A_c^i\mathbb{K}_{ik}^cA_c^k+\alpha_d\sum_{k=1}^{3}A_d^j\mathbb{K}_{jk}^dA_d^k.\end{aligned} \tag{3.30}$$

Replacing the non-zero entries of matrix $\mathbb{N}$ allows to express the right-hand side as

$$\left(\mathbb{N}\mathbf{1}_{\mathsf{F}_p}\right)_f=\alpha_cA_c^i\mathbb{K}_{ii}^cA_c^i+\alpha_dA_d^j\mathbb{K}_{jj}^dA_d^j+\sum_{k=1,k\neq i}^{3}\alpha_cA_c^i\mathbb{K}_{ik}^cA_c^k+\sum_{k=1,k\neq j}^{3}\alpha_dA_d^j\mathbb{K}_{jk}^dA_d^k.$$

Gathering the common terms in the above equation yields

$$\left(\mathbb{N}\mathbf{1}_{\mathsf{F}_p}\right)_f=\alpha_c\sum_{k=1}^{3}A_c^i\mathbb{K}_{ik}^cA_c^k+\alpha_d\sum_{k=1}^{3}A_d^j\mathbb{K}_{jk}^dA_d^k. \tag{3.31}$$

The comparison between (3.30) and (3.31) shows that for all $f\in\mathcal{F}(p)$, $\left(\mathbb{N}^{-1}\mathbb{S}\right)\mathbf{1}_{\mathsf{C}_p}=\mathbf{1}_{\mathsf{F}_p}$, which ends the proof.

### 3.7 Local diffusion matrix at a generic point

In this paragraph, we achieve the space discretization of the diffusion equation gathering the results obtained in the previous sections. We start by recalling the semi-discrete version of the diffusion equation (2.5)

$$m_c C_{vc} \frac{\mathrm{d}}{\mathrm{d}t} T_c + \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{F}(p,c)} A_{pc}^f q_{pc}^f = m_c r_c.$$

We define the contribution of the sub-cell $\omega_{pc}$ to the diffusion flux as

$$Q_{pc} = \sum_{f \in \mathcal{F}(p,c)} A_{pc}^f q_{pc}^f.$$

Using the local numbering of the sub-faces surrounding point $p$ yields to rewrite the above expression as

$$Q_{pc} = \sum_{k=1}^{3} A_c^k q_c^k.$$

Now, we replace the normal flux by its corresponding expression (3.26) to get

$$Q_{pc} = -\sum_{k=1}^{3} A_c^k \left[ \alpha_c \sum_{i=1}^{3} \mathbb{K}_{ki}^c A_c^i (\bar{T}_c^i - T_c) \right].$$

Interchanging the order of the summations in the right-hand side yields

$$Q_{pc} = -\sum_{i=1}^{3} \left[ \alpha_c \sum_{k=1}^{3} (A_c^i \mathbb{K}_{ki}^c A_c^k) \right] (\bar{T}_c^i - T_c).$$

To obtain a more compact form of $Q_{pc}$, we define the matrix $\widetilde{\mathbb{S}}$ whose entries write $\widetilde{\mathbb{S}}_{fc} = \alpha_c \sum_{k=1}^{3} (A_c^i \mathbb{K}_{ki}^c A_c^k)$, where $f \equiv (c,i)$. Employing this notation, the sub-cell contribution to the diffusion flux reads

$$Q_{pc} = -\sum_{f \in \mathcal{F}(p)} \widetilde{\mathbb{S}}_{cf}^t (\bar{T}^f - T_c).$$

Eliminating the sub-face temperatures by means of (3.28) and using the property (3.29) leads to

$$Q_{pc} = -\sum_{d \in \mathcal{C}(p)} \mathbb{G}_{cd}^p (T_d - T_c), \tag{3.32}$$

where $\mathbb{G}^p$ is a $\mathsf{C}_p \times \mathsf{C}_p$ matrix defined at point $p$ by

$$\mathbb{G}^p = \widetilde{\mathbb{S}}^t \mathbb{N}^{-1} \mathbb{S}. \tag{3.33}$$

Let us point out that the entries of $\mathbb{G}^p$ have the physical dimension of a conductivity. Thus, it can be viewed as the effective conductivity tensor at point $p$. More precisely, it

follows from (3.32) that the entry $\mathbb{G}_{cd}^{p}$ stands for the effective conductivity between cells $c$ and $d$ through the point $p$. This node-base effective conductivity tensor will be the cornerstone to assemble the global diffusion matrix over the computational grid.

**Comment 3.6.** If the conductivity tensor $\mathbb{K}$ is symmetric, it is straightforward to show that $\widetilde{\mathbb{S}}=\mathbb{S}$. Bearing this in mind, we claim that $\mathbb{G}^{p}$ is symmetric positive definite provided that the conductivity tensor $\mathbb{K}$ is itself symmetric positive definite. To prove this result, it is sufficient to observe that

$$\begin{aligned}\mathbb{G}^{p}\boldsymbol{T}\cdot\boldsymbol{T}&=(\widetilde{\mathbb{S}}^{t}\mathbb{N}^{-1}\mathbb{S})\boldsymbol{T}\cdot\boldsymbol{T}\\&=\mathbb{N}^{-1}(\mathbb{S}\boldsymbol{T})\cdot(\widetilde{\mathbb{S}}\boldsymbol{T}),\end{aligned}$$

where $\boldsymbol{T}\in\mathfrak{R}^{\mathsf{C}_p}$ is the vector of cell temperatures. Since $\mathbb{K}$ is symmetric, one deduces that $\widetilde{\mathbb{S}}=\mathbb{S}$, in addition $\mathbb{N}$ is symmetric positive definite, which ends the proof.

### 3.8 Assembling of the global diffusion matrix

Taking into account the previous results, the semi-discrete scheme over cell c reads

$$m_c C_{vc}\frac{\mathrm{d}}{\mathrm{d}t}T_c-\sum_{p\in\mathcal{P}(c)}\sum_{d\in\mathcal{C}(p)}\mathbb{G}_{cd}^{p}(T_d-T_c)=m_c r_c, \tag{3.34}$$

where $\mathcal{P}(c)$ is the set of points of cell $c$ and $\mathcal{C}(c)$ is the set of cells surrounding the point $p$. This equation allows to construct the generic entries of the global diffusion matrix, $\mathbb{D}$, as follows

$$\mathbb{D}_{cc}=\sum_{p\in\mathcal{P}(c)}\sum_{d\in\mathcal{C}(p)}\mathbb{G}_{cd}^{p}, \tag{3.35a}$$

$$\mathbb{D}_{cd}=-\sum_{p\in\mathcal{P}(c)}\mathbb{G}_{cd}^{p}, c\neq d. \tag{3.35b}$$

If $\mathsf{C}_{\mathcal{D}}$ denotes the total number of cells composing the computational grid, then matrix $\mathbb{D}$ is a $\mathsf{C}_{\mathcal{D}}\times\mathsf{C}_{\mathcal{D}}$ square matrix. The vector of cell-centered temperatures, $\boldsymbol{\mathcal{T}}\in\mathfrak{R}^{\mathsf{C}_{\mathcal{D}}}$, is the solution of the system of differential equations

$$\mathbb{M}\mathbb{C}_v\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\mathcal{T}}+\mathbb{D}\boldsymbol{\mathcal{T}}=\mathbb{M}\boldsymbol{\mathcal{R}}. \tag{3.36}$$

Here, $\boldsymbol{\mathcal{R}}\in\mathfrak{R}^{\mathsf{C}_{\mathcal{D}}}$ is the source term vector, $\mathbb{M}$ and $\mathbb{C}_v$ are the diagonal matrices whose entries are respectively the cell mass $m_c$ and the cell heat capacity $C_{vc}$.

# 4 Time discretization

In this section, we briefly describe the time discretization of the system (A.10). We restrict the presentation to the case of a linear heat equation knowing that in the non linear case the interested reader might refer to [28]. First, let us prescribe the initial condition $\boldsymbol{\mathcal{T}}(0)=\boldsymbol{\mathcal{T}}^0$, where $\boldsymbol{\mathcal{T}}^0$ is the vector of the cell-averaged initial condition. We solve the system over the time interval $[0,\mathfrak{T}]$ using the subdivision

$$0=t^0<t^1<\cdots<t^n<t^{n+1}<\cdots<t^N=\mathfrak{T}.$$

The time step is denoted by $\Delta t^n=t^{n+1}-t^n$. The time approximation of a quantity at time $t^n$ is denoted using the superscript $n$, for instance $\boldsymbol{\mathcal{T}}^n=\boldsymbol{\mathcal{T}}(t^n)$. Knowing that an explicit time discretization of the diffusion operator necessitates a stability constraint on the time step which is quadratic with respect to the smallest cell size, we prefer to use an implicit time discretization. Further, we assume that the heat capacity and the conductivity tensor do not depend on temperature. Integrating (A.10) over $[t^n,t^{n+1}]$ yields the first-order in time implicit discrete scheme

$$\mathbb{MC}_v\frac{\boldsymbol{\mathcal{T}}^{n+1}-\boldsymbol{\mathcal{T}}^n}{\Delta t^n}+\mathbb{D}\boldsymbol{\mathcal{T}}^{n+1}=\mathbb{M}\boldsymbol{\mathcal{R}}^n+\boldsymbol{\Sigma}^n. \tag{4.1}$$

The updated cell-centered temperatures are obtained by solving the following linear system

$$\left(\frac{\mathbb{MC}_v}{\Delta t^n}+\mathbb{D}\right)\boldsymbol{\mathcal{T}}^{n+1}=\frac{\mathbb{MC}_v}{\Delta t^n}\boldsymbol{\mathcal{T}}^n+\mathbb{M}\boldsymbol{\mathcal{R}}^n+\boldsymbol{\Sigma}^n. \tag{4.2}$$

Let us recall that $\mathbb{D}$ is positive semi-definite. Knowing that $\mathbb{MC}_v$ is a positive diagonal matrix, we deduce that the matrix $\frac{\mathbb{MC}_v}{\Delta t^n}+\mathbb{D}$ is positive definite. Thus, the linear system (4.2) always admits a unique solution. Finally, in the absence of source term and assuming periodic or homogeneous boundary conditions, we observe that the above implicit time discretization is stable with respect to the discrete weighted $L^2$ norm defined by

$$\|\boldsymbol{\mathcal{T}}\|_{w2}^2=(\mathbb{MC}\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}}),$$

where $\boldsymbol{\mathcal{T}}$ is a vector of size $\mathsf{C}_{\mathcal{D}}$. To prove this result, we dot-multiply (4.2) by $\boldsymbol{\mathcal{T}}^{n+1}$ and obtain

$$\mathbb{MC}_v\boldsymbol{\mathcal{T}}^{n+1}\cdot\boldsymbol{\mathcal{T}}^{n+1}-\mathbb{MC}_v\boldsymbol{\mathcal{T}}^n\cdot\boldsymbol{\mathcal{T}}^{n+1}=-\Delta t^n\mathbb{D}\boldsymbol{\mathcal{T}}^{n+1}\cdot\boldsymbol{\mathcal{T}}^{n+1}.$$

Due to the positive definiteness of matrix $\mathbb{D}$ the right-hand side of the above equation is negative, hence

$$\mathbb{MC}_v\boldsymbol{\mathcal{T}}^{n+1}\cdot\boldsymbol{\mathcal{T}}^{n+1}\leq\mathbb{MC}_v\boldsymbol{\mathcal{T}}^n\cdot\boldsymbol{\mathcal{T}}^{n+1}.$$

Employing Cauchy-Schwarz inequality in the right-hand side of the above inequality yields

$$\mathbb{MC}_v\boldsymbol{\mathcal{T}}^n\cdot\boldsymbol{\mathcal{T}}^{n+1}\leq\|\boldsymbol{\mathcal{T}}^n\|_{w2}\|\boldsymbol{\mathcal{T}}^{n+1}\|_{w2}.$$

Gathering the above results leads to

$$\|\boldsymbol{\mathcal{T}}^{n+1}\|_{w2} \leq \|\boldsymbol{\mathcal{T}}^{n}\|_{w2},$$

which ends the proof.

**Comment 4.1.** The computation of the numerical solution requires to solve the sparse linear system (4.2). This is achieved by employing the localized ILU(0) Preconditioned BiCGStab algorithm, refer to [30, 39]. The parallel implementation of this algorithm and its efficiency are discussed in Section 5. Knowing that the matrices encountered in this work are all symmetric, we could have employed a classical conjugate gradient method to solve the corresponding linear system. However, our numerical scheme being able to cope with non-symmetric diffusion equations, refer to [28], we have chosen to implement a more general solver to handle these problems.

# 5 Parallelization

When dealing with three-dimensional grids, the computational power needed to solve the problems grows quickly. In fact two problems occur, the memory consumption becomes higher and the computational time becomes longer. These two problems can be overcome with the parallelization of the scheme. The goal is to split the global problem into smaller problems that will run concurrently on different processors. In the distributed memory case, the more processors we add, the more memory we get. On the other hand communications are then needed between the processors to solve the global problem.

First, we have a look at the implementation of the sequential algorithm and identify the parts we need to parallelize in priority. Then, we describe the partitioning step and the communication process. Finally, we present an experimental study to assess the efficiency of our parallelization scheme.

## 5.1 Analysis of the problem

The sequential algorithm can be divided in two steps: assembling the matrix and solving the system.

To build the global matrix we have to solve a local linear system at each vertex of the mesh. This is a vertex-centered approach. The solving step is performed through the use of iterative Krylov methods such as BiCGStab [39]. These methods need to perform matrix-vector multiplications and dot products. Here, the matrix involved associates a cell with its neighboring cells; hence the solving step is a cell-centered approach.

In a parallel computation we would like to split the problem into equally balanced sub-problems, this is called partitioning. The problem we have to face with our algorithm is that the optimal partition for the vertex-centered approach is different from the optimal

partition for the cell-centered one. We thus have to make a choice, optimizing one step while sacrificing the other.

If we have a look at the sequential timings we can see that the construction process takes approximately 10% of the overall time and the solving step takes 90% of the time. The Amdahl's law [8] tells us that we have more to gain by optimizing the more time consuming step, in our case the solving step, so we will focus on a cell-centered partitioning.

### 5.2 Partitioning and communications

The main kernel of iterative Krylov methods is a matrix-vector multiplication. This is why we have to efficiently parallelize the matrix-vector product $\boldsymbol{Y}=\mathbb{A}\boldsymbol{X}$. We assume that we have a partitioning of our problem, it means that every processor owns a specific subset of the global problem. If $I$ is a processor it will only know the subset $\boldsymbol{X}_I$ of the vector $\boldsymbol{X}$ and the subset $\boldsymbol{Y}_I$ of the vector $\boldsymbol{Y}$. This results in the following decomposition for the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{Y}_1\\ \vdots\\ \boldsymbol{Y}_I\\ \vdots\\ \boldsymbol{Y}_N\end{bmatrix},\quad \boldsymbol{X}=\begin{bmatrix}\boldsymbol{X}_1\\ \vdots\\ \boldsymbol{X}_I\\ \vdots\\ \boldsymbol{X}_N\end{bmatrix}.$$

Similarly, we decompose the matrix $\mathbb{A}$ and express it in terms of a block matrix with $\mathbb{A}_{IJ}$ elements:

$$\mathbb{A}=\begin{bmatrix}\mathbb{A}_{11} & \cdots & \mathbb{A}_{1N}\\ & \ddots & \\ \vdots & \mathbb{A}_{IJ} & \vdots\\ & & \ddots\\ \mathbb{A}_{N1} & \cdots & \mathbb{A}_{NN}\end{bmatrix}.$$

With these notations the matrix-vector multiplication $\boldsymbol{Y}=\mathbb{A}\boldsymbol{X}$ may be expressed as:

$$\boldsymbol{Y}_I=\sum_{J=1,\cdots,N}\mathbb{A}_{IJ}\boldsymbol{X}_J,\quad \forall I\in\{1,\cdots,N\}.$$

To compute subvector $\boldsymbol{Y}_I$ processor $I$ needs to access the bloc matrices $\mathbb{A}_{IJ}$ where $J=1,\cdots,N$. More precisely, processor $I$ needs to access all the rows associated to its partition. We say that the matrix is partitioned row-wise. Processor $I$ also potentially needs to know the vectors $\boldsymbol{X}_J$ where $J=1,\cdots,N$, which is the whole $\boldsymbol{X}$ vector. As we mentioned before, it only owns $\boldsymbol{X}_I$ vector. To perform the global operation, we thus need to receive the subvectors $\boldsymbol{X}_J$ from the processors $J$ ($J\neq I$). As matrix $\mathbb{A}$ is sparse, a block $\boldsymbol{X}_J$ is effectively needed if and only if $\mathbb{A}_{IJ}$ has non zero entries. Furthermore, $\mathbb{A}_{IJ}$ may also be

sparse. Thus, only part of the elements of subvector $\boldsymbol{X}_J$ may be needed on processor $I$. We note $\hat{\boldsymbol{X}}_J^I$ the corresponding pruned subvector (so-called "overlap") and $\boldsymbol{Y}_I \leftarrow \mathbb{A}_{IJ}\boldsymbol{X}_J$ may be compacted into $\boldsymbol{Y}_I \leftarrow \hat{\mathbb{A}}_{IJ}\hat{\boldsymbol{X}}_J^I$.

The algorithm for the parallel matrix-vector on processor $I$ can then be written as follows:

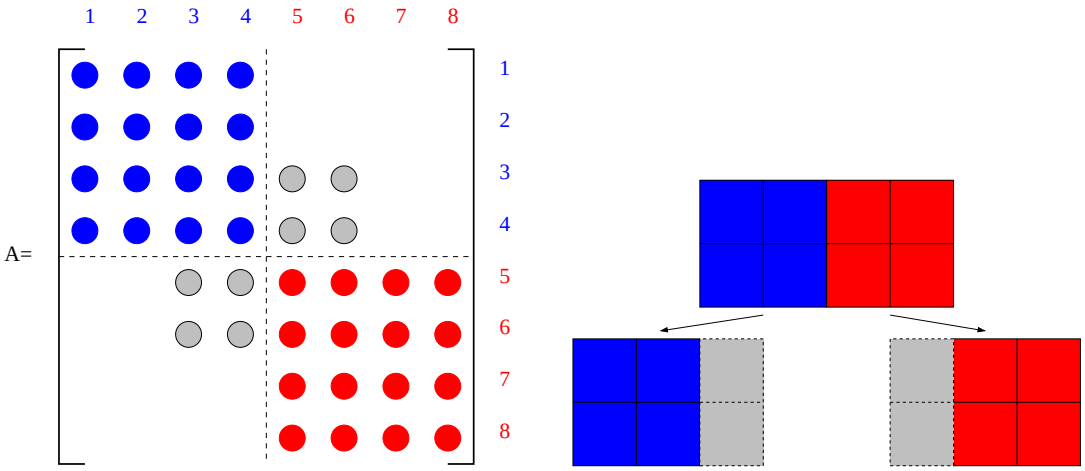
- For each processor $J$, send $\hat{\boldsymbol{X}}_I^J$ to processor J.
- Compute $\boldsymbol{Y}_I \leftarrow \mathbb{A}_{II}\boldsymbol{X}_I$.
- For each processor $J$, receive $\hat{\boldsymbol{X}}_J^I$ from processor $J$ and compute $\boldsymbol{Y}_I \leftarrow \boldsymbol{Y}_I + \hat{\mathbb{A}}_{IJ}\hat{\boldsymbol{X}}_J^I$.

In order to hide the communication process, we consider non-blocking communications, which occur while we compute the local matrix-vector multiplication $\boldsymbol{Y}_I \leftarrow \mathbb{A}_{II}\boldsymbol{X}_I$. If the amount of computation for this operation is big enough the distant subvector $\hat{\boldsymbol{X}}_J^I$ may be transferred without impact on the elapsed time.

A simple example of this decomposition is displayed in Fig. 6(a). In this example, the first processor in blue owns the first four rows of the matrix and the first six elements of the vector, while the second processor in red owns the last four rows and the last six elements of the vector. On the first processor the overlaps are the element 5 and 6 of the vector, while the overlaps of the second processor consist of the elements 3 and 4. These elements are not computed on the local processor but are received from the other processor. The associated mesh and the corresponding sub-meshes obtained after the decomposition are shown in Fig. 6(b), the grey cells represent the overlaps. The other parallel operation to perform in our iterative solver is the inner product $p=\boldsymbol{X}\cdot\boldsymbol{Y}=\sum_k \boldsymbol{X}_k\boldsymbol{Y}_k$. With our decomposition it writes $p=\sum_{J=1,\cdots,N} p_J$ where $p_J=\sum_k \boldsymbol{X}_J^k\boldsymbol{Y}_J^k$ is the local inner product corresponding to the sub-problem $J$. This is a global operation, every processor has to compute its local inner product and exchange it with all the other processors. This communication cannot be overlapped by computation, this could be a bottleneck in our algorithms.

How should we distribute the rows of the matrix? As we said before we want the load to be balanced between the processors, in our case it is the number of operations in the matrix-vector operation or the number of non-zero elements in the matrix. We also want to overlap the communications with computations, and the communication time depends on the amount of data to exchange. This amounts to reduce the volume of communication between processors. This problem is really complicated, that is why to achieve these goals we use a graph partitioner called Scotch [33]. We process the graph associated to the global matrix with this library which retrieves for each row the partition it belongs to. With these informations we can set up the communication scheme explained earlier.

What are the changes needed by the scheme in parallel? When we add an element in the overlap vector we add the corresponding cell in the local sub-mesh. So in every sub-mesh an internal cell is surrounded by all its neighbors, we have all the information




(a) Matrix decomposition (b) Mesh decomposition

Figure 6: Example of matrix and mesh decomposition on two processors with overlaps construction.

to build the row corresponding to this cell in the matrix, so nothing has to be changed in the scheme. The parallelism is only seen in the solving step.

We have implemented these methods in our development code but we can also use the PetSC library [9–11]. This library implements scalable algorithms to solve scientific applications modeled by partial differential equations. In this library the solving step and the communication process are hidden to the user. The problem we faced is that with the libraries available on our experimental platform we could not use some preconditioners in parallel like ILU(0). Due to that, the iterative method does not converge very well in parallel. This explains why the experiments we ran with PetSC were not very conclusive.

### 5.3 Experiments

In order to quantify the quality of the parallelization we define two metrics: the speedup and the efficiency. If $T_p$ denotes the time needed to solve the problem on $p$ processors the speedup is defined by

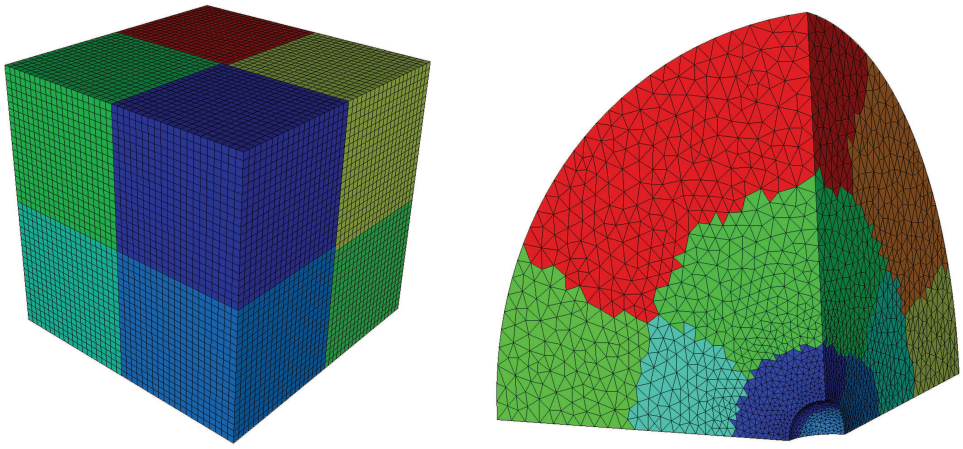
$$S(p)=\frac{T_1}{T_p}.$$

This quantity represents how much faster the algorithm is on $p$ processors than in sequential. Ideally on $p$ processors we would like to be $p$ times faster than in sequential, thus the ideal speedup is defined by

$$S_{ideal}(p)=p.$$

An other interesting metric is the efficiency. It is given by

$$E(p)=\frac{T_1}{pT_p}.$$



(a) Structured hexahedral grid (b) Unstructured tetrahedral grid

Figure 7: Partitioning computed with the Scotch library (INRIA) [33] for 8 processors. One distinct color is attributed to each processor.

It assesses how efficiently the processors are used with respect to the ideal case ($E(p)=1$).

We ran the experiments on the PLAFRIM (IMB/LABRI/INRIA) [1] platform. On each node of this machine we have 2 Quad-core Nehalem Intel Xeon X5550 (8 CPU cores total per node) running at 2,66 GHz. The nodes have 24Gb of RAM (DDR3 1333MHz) and are connected with Infiniband QDR at 40Gb/s. To test the scalability of our method we ran the tests on 1 to 64 CPU cores (using 1 to 8 nodes). We run the speedup tests on two kinds of grids:

- a Cartesian hexahedral grid made of 512 000 cells, 531 441 nodes and with 13 481 272 non-zeros entries in the associated matrix;
- a unstructured tetrahedral grid made of 396 601 cells, 98 218 nodes and with 28 946 047 non-zeros entries in the associated matrix.

We specifically chose these meshes to illustrate the load balancing problem occurring between the matrix construction and the solving step. On the first kind of mesh we have a perfect load balancing while on the second one the load balancing of the construction step can be bad, due to the unstructured feature of the grid. The partitioning of the coarsest versions of these grids are displayed in Fig. 7(a) for the structured grid and in Fig. 7(b) for the unstructured grid. On the speedup curve displayed in Fig. 8 we can see that the more processors we add, the further away from the ideal speedup we get. This highlights two different phenomena. First, in the conjugate gradient method we need to compute some scalar products and vector norms which need collective communications. This kind of communications does not scale very well so the more processors we add the worse it gets. The other phenomenon is that by adding more processors, the local matrices get smaller and we need to communicate more at the same time. So at some

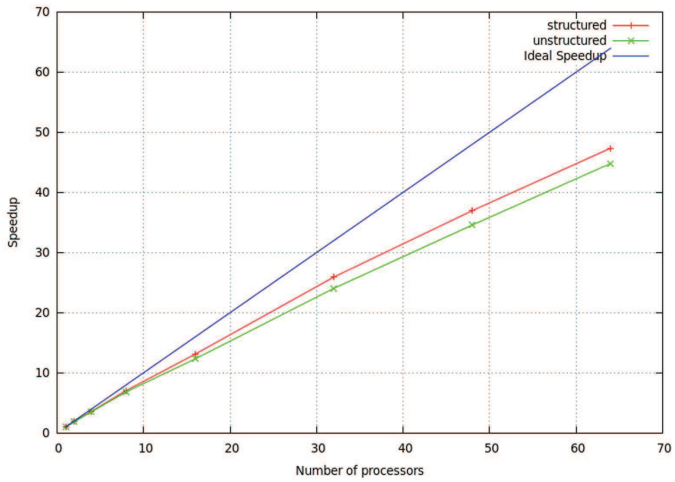



Figure 8: Speedup curve for 1 to 64 processors on structured and unstructured 3D meshes.

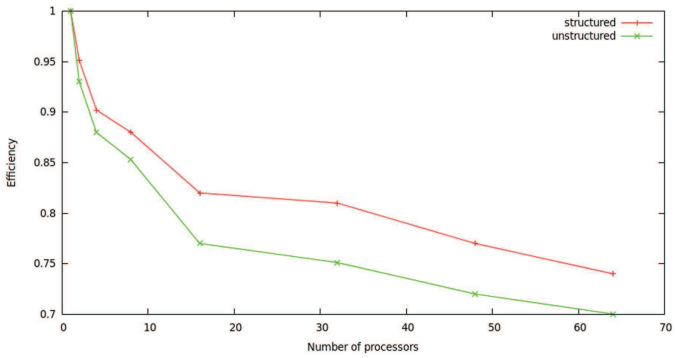



Figure 9: Efficiency curve for 1 to 64 processors on structured and unstructured three-dimensional meshes.

point the computation can not overlap the communications anymore and the speedup gets worse.

On the efficiency curve displayed in Fig. 9 we can see that from 1 to 8 CPU cores the efficiency quickly drops from 1 to 0.85, then between 8 to 64 CPU cores it decreases more slowly. This reflect the topology of the platform we used for the tests. From 1 to 8 CPU cores we are only using one node. On a single node the communication cost is negligible, so we would expect the efficiency to stay close to 1. The quick drop shows the existence of a bottleneck in the memory usage. This may come from the usage of unstructured methods which uses a lots of memory indirection, some optimization around the matrix numbering should reduce this effect. The decrease in efficiency observed with more than 8 CPU cores is due to the communications between the nodes.

Finally, we can comment the difference between the efficiency obtained on structured and unstructured meshes. We observe the same phenomenon on the two kinds

of meshes. We observe that the efficiency is a bit better for the structured meshes. The difference is due to the imbalance in the construction step. This imbalance could be lowered by adding information about the cost of the matrix construction into the graph sent to the partitioner.

To conclude this paragraph we can claim that we developed a parallel implementation of the classical BiCGStab method. This method has some blocking points, the inner products that can not be overlapped by computations, so it is not fully scalable. In [41] the authors present the IBiCGStab method, a modified BiCGStab algorithm with an equivalent numerical stability, in which these blocking points are cured. Only one global synchronization point is needed per iteration instead of four in the original algorithm. This is a more scalable method. In a future work we plan to investigate this modification to improve the efficiency of our implementation.

# 6 Numerical results

The aim of this section is to assess the robustness and the accuracy of our finite volume scheme against analytical test cases using various types of unstructured three-dimensional grids. The tests cases have been chosen to highlight the different features of the scheme. First, we describe the methodology employed for the convergence analysis defining the related metrics. Then, we present the three-dimensional structured and unstructured grids employed. Finally, we describe the set up of each test case, display the numerical results obtained and discuss the quality of the corresponding convergence analysis.

In this section, the numerical solutions are obtained solving linear systems by means of the localized ILU(0) Preconditioned BiCGStab algorithm [30, 39]. The relative error tolerance to achieve the convergence is equal to $10^{-16}$.

## 6.1 Methodology

Let us recall that we are solving the generic diffusion equation

$$\rho C_v \frac{\partial T}{\partial t} - \nabla \cdot (\mathbb{K} \nabla T) = \rho r, \qquad (\boldsymbol{x},t) \in \mathcal{D} \times [0,\mathcal{T}], \tag{6.1a}$$

$$T(\boldsymbol{x},0) = T^0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{D}, \tag{6.1b}$$

where $r = r(\boldsymbol{x})$ is a source term. The analytical solutions of all the tests are stationary. Thus, we are going to compute them starting with the initial condition $T^0(\boldsymbol{x}) = 0$ and we run the simulation until the steady state is reached. The density and the specific heat capacity are specified such that $\rho = 1$ and $C_v = 1$. The boundary conditions, the source term and the heat conductivity tensor $\mathbb{K}$ will be specified for each test case.

Bearing this in mind, we describe the procedure employed to perform the conver-

gence analysis. First, we define the mesh resolution

$$h=\left(\frac{|\mathcal{D}|}{\mathsf{C}_{\mathcal{D}}}\right)^{\frac{1}{d}},$$

where $\mathsf{C}_{\mathcal{D}}$ denotes the number of cells that paved the computational domain and $d=3$ is the dimension of the space. Let $T=\hat{T}(\boldsymbol{x})$ be the steady analytical solution of the diffusion equation (6.1a). Being given a computational grid characterized by $h$, we denote by $\hat{T}_c^h$ the value of the analytical solution evaluated at the centroid of the cell $\omega_c$, *i.e.*, $\hat{T}_c^h=\hat{T}(\boldsymbol{x}_c)$, where $\boldsymbol{x}_c$ is the cell centroid. If $T_c^h$ denotes the cell averaged temperature computed by the numerical scheme, we define the asymptotic numerical errors based on the discrete $L^2$ and $L^\infty$ norms

$$E_2^h=\sqrt{\sum_{c=1}^{\mathsf{C}_{\mathcal{D}}}(T_c^h-\hat{T}_c^h)^2\,|\,\omega_c\,|},$$
$$E_\infty^h=\max_{c=1\cdots\mathsf{C}_{\mathcal{D}}}|\,T_c^h-\hat{T}_c^h\,|.$$

The asymptotic error for both norms is estimated by

$$E_\alpha^h=C_\alpha h^{q_\alpha}+\mathcal{O}(h^{q_\alpha+1})\quad\text{for}\quad\alpha=2,\infty. \tag{6.2}$$

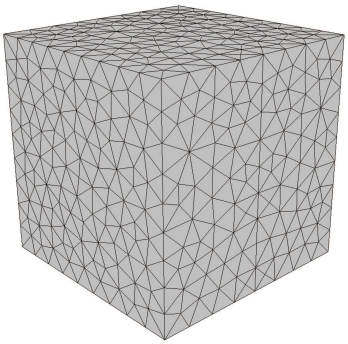
Here, $q_\alpha$ denotes the order of the truncation error and $C_\alpha$ is the convergence rate-constant which is independent of $h$. Having computed the asymptotic errors corresponding to two different grids characterized by mesh resolutions $h_1$ and $h_2<h_1$, we deduce an estimation of the order of truncation error as

$$q_\alpha=\frac{\log E_\alpha^{h_2}-\log E_\alpha^{h_1}}{\log h_2-\log h_1}. \tag{6.3}$$

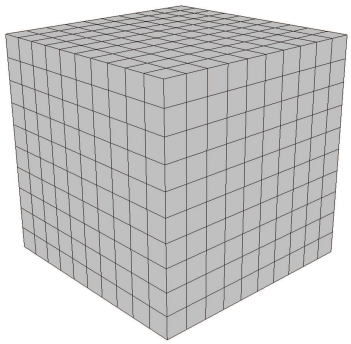
## 6.2 Computational grids

Here, we present the three-dimensional computational grids employed to run the test cases. There are various types of grids: tetrahedral grids, hexahedral grids and hybrid grids which are composed of tetrahedra, hexahedra and pyramids. The detailed description of these grids is summarized in the list below:

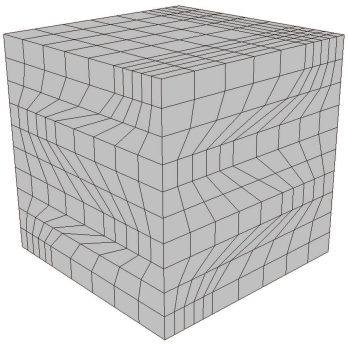
- Tetrahedral grids, displayed in Fig. 10(a) and Fig. 10(f), have been constructed using Gmsh, which is a three-dimensional finite element mesh generator [17];
- Hexahedral Cartesian grid displayed in Fig. 10(b);
- Kershaw-type grid displayed in Fig. 10(c);



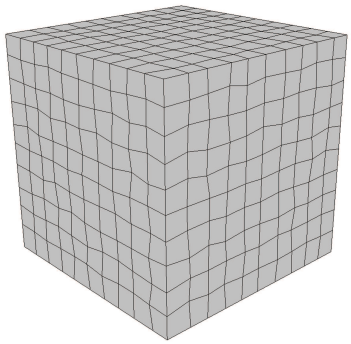
(a) Tetrahedral grid made of 8222 cells



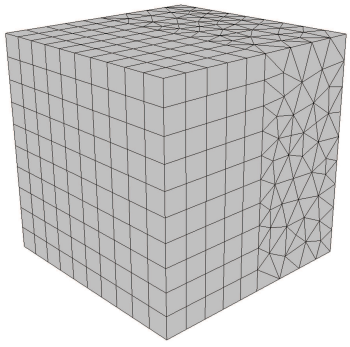
(b) Cartesian hexahedral grid made of 1000 cells



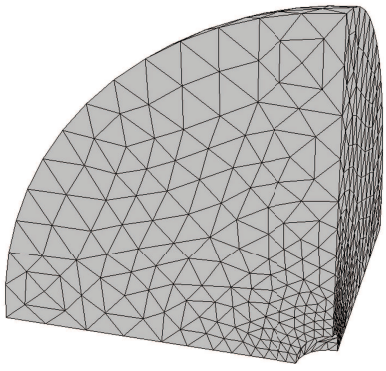
(c) Kershaw-type grid made of 1000 cells



(d) Randomly perturbed grid made of 1000 cells



(e) Hybrid grid made of 3432 tetrahedra, 100 pyramids and 500 hexahedra



(f) Tetrahedral grid of a truncated sphere made of 6623 cells

Figure 10: Three-dimensional grids used for the test cases.

- Smoothly deformed hexahedral grid resulting from the mapping defined on the unit cube $[0,1]^3$ by

$$\begin{aligned} x(\xi,\eta,\theta) &= \xi + a_0 \sin(2\pi\xi)\sin(2\pi\eta)\sin(2\pi\theta), \\ y(\xi,\eta,\theta) &= \eta + a_0 \sin(2\pi\xi)\sin(2\pi\eta)\sin(2\pi\theta), \\ z(\xi,\eta,\theta) &= \theta + a_0 \sin(2\pi\xi)\sin(2\pi\eta)\sin(2\pi\theta), \end{aligned}$$

  where the amplitude of the deformation is $a_0 = 0.1$. Observing that the deformation cancels on the boundary surfaces of the unit cube, this grid has not been displayed since it looks like the Cartesian grid;

- Randomly deformed hexahedral grid, displayed in Fig. 10(d), resulting from the mapping defined on the unit cube $[0,1]^3$ by:

$$\begin{aligned} x(\xi,\eta,\theta) &= \xi + a_0 h r_1, \\ y(\xi,\eta,\theta) &= \eta + a_0 h r_2, \\ z(\xi,\eta,\theta) &= \theta + a_0 h r_3, \end{aligned}$$

  where $\{r_i\}_{i=1,\cdots,3}$ are random numbers in $[-1,1]$, $h$ is the characteristic mesh size and $a_0 = 0.2$ the amplitude of the deformation;

- Hybrid grid, displayed in Fig. 10(e), made of hexahedral cells, pyramidal cells and tetrahedra.

**Comment 6.1.** We have introduced the hybrid grid because of its usefulness regarding real-world applications. Let us point out that it is a convenient way to mesh a domain using both hexahedral and tetrahedra cells with the constraint of keeping a conformal grid. In this case, a layer of pyramids ensures the transition between hexahedra and tetrahedra. This kind of grid can be used in the context of the computation of a viscous flow in the presence of a solid wall. Indeed, the pyramid cells allows to match the boundary layer in the vicinity of the wall, paved by means of hexahedra, with the rest of the domain paved using tetrahedra.

**Comment 6.2.** The tetrahedral grid displayed in Fig. 10(f) corresponds to a truncated sphere with an internal radius $R_i$ and an external radius $R_e$. This grid is also characterized by an interface located at $R_m$, which allows to separate two distinct materials.

### 6.3 Isotropic diffusion problem

This problem consists in finding the steady solution of (6.1) with $r = 0$ and an isotropic conductivity tensor defined by $\mathbb{K} = \kappa\mathbb{I}$, where $\mathbb{I}$ is the unit tensor of $\mathfrak{R}^3$ and the scalar conductivity is given by $\kappa = 1$. The computation domain is $\mathcal{D} = [0,1]^3$ and we apply the following boundary conditions on the boundaries of $\mathcal{D}$

- Dirichlet boundary condition

$$T(\boldsymbol{x})=0, \quad \text{for } x=0,$$
$$T(\boldsymbol{x})=1, \quad \text{for } x=1.$$

- Neumann boundary condition

$$\boldsymbol{q}(x)\cdot\boldsymbol{n}=0, \quad \text{for } y=0,\ y=1,\ z=0 \quad \text{and} \quad z=1.$$

The steady analytical solution is $\hat{T}(\boldsymbol{x})=x$. The aim of this simple test case is to assess the ability of our scheme to preserve linear fields.

First, we compute the steady numerical solution using a tetrahedral grid made of 8222 cells, refer to Fig. 10(a). The corresponding asymptotic errors are equal to zero up to machine precision. As expected, our finite volume scheme preserves linear solutions on tetrahedral grids. We observe a similar behavior when computing the numerical solution on the Cartesian hexahedral grid displayed in Fig. 10(b). Let us point that this result confirms the conclusion already drawn for this type of numerical methods, in the context of two-dimensional geometry, refer to [13,28]. The convergence analysis for smooth grids, Kershaw grids (refer to Fig. 10(c)) and random grids (refer to Fig. 10(d)) are performed computing the asymptotic errors and the corresponding orders of truncation error using formulas (6.2) and (6.3). The results displayed in Table 1(a) show that the convergence rate is almost of second-order in the $L^2$ norm and a little bit less in the $L^\infty$ norm. In Table 1(b), we observe a similar behavior for the convergence analysis corresponding to the Kershaw grids. Proceeding with the convergence analysis for random grids as before, we have displayed the corresponding results in Table 1(c). The convergence rate is of first-order for the $L^2$ norm and almost of first-order for the $L^\infty$ norm.

**Comment 6.3.** Let us point out that the last columns in the above tables represent the number of iterations required by our ILU(0) Preconditioned BiCGStab algorithm to reach the relative error tolerance required to achieve the convergence in solving the linear systems. This error tolerance has been set equal to $10^{-16}$ which is a very small tolerance. This probably explains the relatively high number of iterations of our solver. It is worth mentioning that for real life applications we shall set the error tolerance equal to $10^{-8}$.

Finally, the convergence analysis for the hybrid grids, refer to Fig. 10(e), is displayed in Table 2. The corresponding data demonstrate that our numerical scheme exhibits a rate of convergence located between first and second-order. Let us point out that the maximal error is always located in the layer of pyramids which allows to link the tetrahedral and the hexahedral regions of the grid. This clearly shows that the loss of accuracy is the consequence of the particular treatment applied to pyramids to derive the flux approximation, refer to Appendix B.

Table 1: Isotropic diffusion problem, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for hexahedral grids.

(a) Smooth grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 5.65D-03 | **1.60** | 2.18D-03 | **1.69** | 11 |
| 5.00D-02 | 1.87D-03 | **1.80** | 6.75D-04 | **1.90** | 22 |
| 2.50D-02 | 5.35D-04 | **1.92** | 1.81D-04 | **1.97** | 44 |
| 1.25D-02 | 1.41D-04 | - | 4.63D-05 | - | 101 |

(b) Kershaw grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 3.22D-02 | **1.94** | 8.23D-03 | **2.04** | 14 |
| 5.00D-02 | 8.39D-03 | **1.39** | 2.00D-03 | **1.69** | 37 |
| 2.50D-02 | 3.20D-03 | **2.09** | 6.20D-04 | **2.06** | 80 |
| 1.25D-02 | 7.53D-04 | - | 1.49D-04 | - | 116 |

(c) Random grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 2.09D-03 | **0.75** | 6.62D-04 | **1.00** | 11 |
| 5.00D-02 | 1.24D-03 | **0.81** | 3.31D-04 | **1.00** | 21 |
| 2.50D-02 | 7.07D-04 | **0.93** | 1.66D-04 | **1.00** | 42 |
| 1.25D-02 | 3.72D-04 | - | 8.29D-05 | - | 78 |

Table 2: Isotropic diffusion problem, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for hybrid grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 6.28D-02 | 2.06D-03 | **0.99** | 2.66D-04 | **1.49** | 14 |
| 3.12D-02 | 1.04D-03 | **1.00** | 9.39D-05 | **1.50** | 27 |
| 1.56D-02 | 5.19D-04 | **1.74** | 3.32D-05 | **2.59** | 60 |
| 1.04D-02 | 2.58D-04 | - | 1.17D-05 | - | 95 |

## 6.4 Isotropic diffusion problem with a discontinuous conductivity

Here, the computational domain, $\mathcal{D}$, is the truncated sphere, centered at the origin and characterized by the inner radius $R_i$=0.1 and the outer radius $R_e$=1. An interface, located at the radius $R_m = 0.5$, splits the computational domain into two regions filled with two distinct materials. The conductivity tensor is isotropic and piecewise constant, *i.e.*, $\mathbb{K}=\kappa\mathbb{I}$

where $\kappa=\kappa(r)$ with $r=\sqrt{x^2+y^2+z^2}$. The scalar conductivity is given by

$$\kappa(r)=\begin{cases}\kappa_1, & \text{if } r\in[R_i,R_m[,\\ \kappa_2, & \text{if } r\in]R_m,R_e].\end{cases}$$

For numerical applications, we choose $\kappa_1=10$ and $\kappa_2=1$. Dirichlet boundary conditions are prescribed at the inner and the outer boundary of the computational domain, *i.e.*, $T(R_i)=T_i=0$ and $T(R_e)=T_e=1$. Due the radial symmetry of the problem, we consider a computational domain restricted to $\frac{1}{8}$ of the truncated sphere. The corresponding coarsest tetrahedral grid is displayed in Fig. 10(f). Homogeneous Neumann boundary conditions are prescribed at the remaining boundaries of the computational domain to handle the symmetry of the problem.

The steady analytical temperature, $\hat{T}(r)$, field is obtained by solving the following problem

$$\begin{aligned}&\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}T}{\mathrm{d}r}\right)=0, \quad r\in]R_i,R_e[,\\ &T(R_i)=T_i,\\ &T(R_m^-)=T(R_m^+), \quad \kappa_1\frac{\mathrm{d}T}{\mathrm{d}r}(R_m^-)=-\kappa_2\frac{\mathrm{d}T}{\mathrm{d}r}(R_m^+),\\ &T(R_e)=T_e.\end{aligned}$$

Let us remark that the second equation in the above system expresses the continuity conditions of the temperature and the heat flux across the interface located at $R_m$. Employing the previous numerical values, the analytical solution reads

$$\hat{T}(r)=\begin{cases}-\frac{1}{18r}+\frac{5}{9}, & \text{if } r\in[R_i,R_m],\\ -\frac{5}{9r}+\frac{14}{9}, & \text{if } r\in[R_m,R_e].\end{cases}$$

The steady analytical and numerical solutions are displayed in Fig. 11. We have plotted the averaged temperature of all the cells versus the cell centroid radius. We observe that the numerical solution is almost superimposed to the analytical solution. **This clearly shows the ability of the scheme to preserve the radial symmetry on a highly anisotropic unstructured grid, which is not aligned with the symmetry of the problem**. We investigate the convergence analysis for this problem using a sequence of four tetrahedral grids made of 902, 6623, 13549 and 37648 cells. The resulting asymptotic errors and rate of convergence in both $L^\infty$ and $L^2$ norms are presented in Table 3. We observe that a second-order rate of convergence is asymptotically reached in $L^2$ norm. It is interesting to note the big gap in the maximum errors between the coarsest grid and the second grid. This might be due to the discretization of the spherical boundaries. On the coarsest grid, the mesh resolution is too poor to properly capture the curvilinear inner and outer boundaries. When the grid is refined, the curvilinear feature of the boundaries is better captured due to the increased mesh resolution.

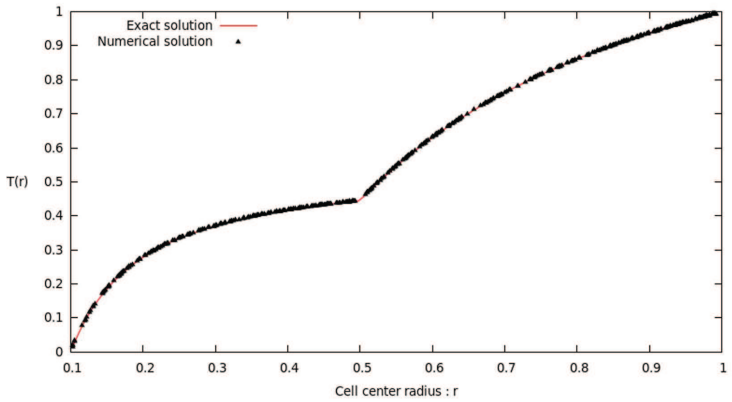



Figure 11: Isotropic diffusion problem with a discontinuous conductivity: Temperatures in all the cells with respect to the radii of the cell centroid for a tetrahedral grid composed of 13549 tetrahedra; comparison with the analytical solution.

Table 3: Isotropic diffusion problem with a discontinuous conductivity, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for tetrahedral grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.20D-01 | 1.23D-01 | **3.19** | 2.16D-02 | **2.49** | 7 |
| 5.80D-02 | 1.22D-02 | **1.65** | 3.57D-03 | **2.25** | 18 |
| 4.54D-02 | 8.14D-03 | **1.90** | 2.05D-03 | **2.08** | 39 |
| 3.17D-02 | 4.12D-03 | - | 9.73D-04 | - | 59 |

## 6.5 Anisotropic diffusion problem with a highly heterogeneous conductivity tensor

This paragraph consists in assessing our finite volume scheme against a test case which is representative of anisotropic diffusion characterized by a highly heterogeneous conductivity tensor. This test case and its manufactured analytical solution are taken from [21]. Here, we solve the problem (6.1) over the computational domain $\mathcal{D}=[0,1]^3$. The conductivity tensor is defined by

$$\mathbb{K}=\mathbb{Q}\begin{pmatrix}1 & 0 & 0\\ 0 & \varepsilon & 0\\ 0 & 0 & \eta(1+x+y+z)\end{pmatrix}\mathbb{Q}^t.$$

where, $\mathbb{Q}=\mathbb{Q}(x)$ is the rotation given by

$$\mathbb{Q}=\begin{pmatrix}\cos(\pi x) & -\sin(\pi x) & 0\\ \sin(\pi x) & \cos(\pi x) & 0\\ 0 & 0 & 1\end{pmatrix}.$$

Here, $\varepsilon$ and $\eta$ are parameters which measure the degree of anisotropy of the conductivity tensor. Indeed, the eigenvalues of the conductivity tensor are: 1, $\varepsilon$ and $\eta(1+x+y+z)$.

Table 4: Anisotropic diffusion problem, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for tetrahedral grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.12D-01 | 3.01D-01 | **2.43** | 7.98D-02 | **2.55** | 10 |
| 4.95D-02 | 4.18D-02 | **1.94** | 1.01D-02 | **2.14** | 14 |
| 2.47D-02 | 1.08D-02 | **1.64** | 2.26D-03 | **1.99** | 32 |
| 1.23D-02 | 3.45D-03 | - | 5.70D-04 | - | 61 |

For numerical applications, we shall take $\varepsilon = 0.1$ and $\eta = 10$. The source term, $r = r(\boldsymbol{x})$, is computed such that the analytical steady solution of (6.1) is given by

$$\hat{T}(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

We apply a homogeneous Dirichlet boundary condition on the boundaries of the computational domain, *i.e.*, $T(\boldsymbol{x},t) = 0$, $\forall\, \boldsymbol{x} \in \partial\mathcal{D}$. First, we compute the numerical solution using a sequence of four tetrahedral grids. The coarsest grid has been displayed in Fig. 10(a). The asymptotic errors in both $L^\infty$ and $L^2$ norms and the corresponding truncation errors are summarized in Table 4. They show that the convergence rate in $L^2$ norm is of second-order. Regarding the convergence analysis on hexahedral grids, we have also used a sequence of four grids for the following types of grids: Cartesian, Kershaw, smooth and random. These grids are showed respectively in Fig. 10(b), Fig. 10(c) and Fig. 10(d). Let us recall that the smooth grid has not been displayed since the deformation cancels on the boundaries of the computational domain. We start by giving in Table 5(a) the convergence analysis data for a sequence of four Cartesian grids. These data demonstrate that our scheme exhibits an almost second-order rate of convergence on Cartesian grids. The same conclusion holds for the convergence analysis performed on smooth grids, refer to Table 5(c). We observe that the rate of convergence in $L^2$ norm are better than those obtained for the rectangular grids, however the asymptotic errors on smooth grids are approximately three times bigger than the ones corresponding to the Cartesian grids. Next, we pursue our investigation using a sequence of four Kershaw grids. The related convergence analysis is summarized in Table 5(b). This time, the rate of convergence in both $L^\infty$ and $L^2$ norms is lying between first and second-order. Finally, we compute the numerical solution on a sequence of four random grids. The results of the convergence analysis corresponding to this sequence of grids are given in Table 5(d). In comparison to the above results, these ones are representative of an erratic behavior, which clearly does not correspond to a second-order rate of convergence.

We achieve the convergence analysis of the present problem by studying the numerical solutions obtained employing a sequence of four hybrid grids, refer to Fig. 10(e). The asymptotic errors and the convergence rates in both $L^\infty$ and $L^2$ norms are displayed in Table 6. The results demonstrate that the scheme is characterized by a rate of convergence located between first and second-order. Once more, the maximal error is located

Table 5: Anisotropic diffusion problem, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for hexahedral grids.

(a) Cartesian grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00e-01 | 1.32D-02 | **1.68** | 4.86D-03 | **1.91** | 10 |
| 5.00D-02 | 4.13D-03 | **1.62** | 1.30D-03 | **1.95** | 21 |
| 2.50D-02 | 1.34D-03 | **1.69** | 3.35D-04 | **1.98** | 43 |
| 1.25D-02 | 4.15D-04 | - | 8.50D-05 | - | 112 |

(b) Kershaw grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 3.01D-02 | **1.15** | 7.80D-03 | **1.72** | 13 |
| 5.00D-02 | 1.36D-02 | **1.25** | 2.38D-03 | **1.55** | 28 |
| 2.50D-02 | 5.72D-03 | **1.65** | 8.13D-04 | **1.99** | 53 |
| 1.25D-02 | 1.82D-03 | - | 2.05D-04 | - | 109 |

(c) Smooth grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 6.03D-02 | **2.09** | 1.60D-02 | **2.12** | 16 |
| 5.00D-02 | 1.41D-02 | **1.68** | 3.69D-03 | **2.03** | 37 |
| 2.50D-02 | 4.41D-03 | **1.89** | 9.03D-04 | **2.00** | 87 |
| 1.25D-02 | 1.19D-03 | - | 2.26D-04 | - | 123 |

(d) Random grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 1.00D-01 | 2.44D-02 | **1.41** | 7.35D-03 | **1.86** | 11 |
| 5.00D-02 | 9.17D-03 | **1.09** | 2.02D-03 | **1.56** | 21 |
| 2.50D-02 | 4.31D-03 | **0.86** | 6.85D-04 | **0.72** | 46 |
| 1.25D-02 | 2.37D-03 | - | 4.16D-04 | - | 94 |

Table 6: Anisotropic diffusion problem, asymptotic errors in both $L^\infty$ and $L^2$ norms and corresponding truncation errors for hybrid grids.

| $h$ | $E_\infty^h$ | $q_\infty^h$ | $E_2^h$ | $q_2^2$ | Iterations |
|---|---|---|---|---|---|
| 6.28D-02 | 4.67D-02 | **1.63** | 9.58D-03 | **2.01** | 18 |
| 3.12D-02 | 1.49D-02 | **1.47** | 2.34D-03 | **1.93** | 26 |
| 1.56D-02 | 5.38D-03 | **1.31** | 6.14D-04 | **1.83** | 65 |
| 1.04D-02 | 3.18D-03 | - | 2.95D-04 | - | 128 |

in the layer of pyramids which allows to link the tetrahedral and the hexahedral regions of the grid. Let us repeat that this loss of accuracy is the consequence of the particular treatment applied to pyramids to derive the flux approximation, refer to Appendix B.

# 7 Conclusion

In this paper, we have described a cell-centered finite volume scheme, which aims at solving anisotropic diffusion problems on three-dimensional unstructured grids. This scheme is characterized by cell-centered unknowns, a local stencil and a symmetric positive definite matrix. The partition of grid cells (resp. faces) into sub-cells (resp. –faces) allows to construct a sub-face fluxes approximation by means of a sub-cell-based variational formulation. The sub-face fluxes are locally expressed at each node in terms of the surrounding cell-centered temperatures invoking continuity conditions of temperature and normal heat flux at each cell interface. Regarding its accuracy, the scheme preserves linear fields with respect to the space variable over tetrahedral grids and exhibits an almost second-order rate of convergence on smooth distorted hexahedral grids. The parallel implementation of the scheme is discussed and its evaluation shows a satisfying efficiency.

In future, we plan to develop an arbitrary Lagrangian Eulerian (ALE) formulation of the present scheme to solve Stefan-like problems, *i.e.*, phase change problems, over a moving grid.

# Acknowledgments

R. Abgrall has been funded in part by the EU ERC Advanced grant "ADDECCO" #226616.

# A Properties of the semi-discrete scheme and practical issues

In this section, we describe briefly some interesting properties that characterize our finite volume semi-discrete scheme. Firstly, we show that the scheme is characterized by a positive semi-definite global diffusion matrix. Secondly, we demonstrate the $L^2$-stability of the space discretization. Then, we present the boundary conditions implementation. Finally, we give some comments about the size of the local node-based sparse linear systems satisfied by the sub-face temperatures.

## A.1 Positive semi-definiteness of the global diffusion matrix

We demonstrate that the global diffusion matrix, $\mathbb{D}$, is positive semi-definite, that is for all $\boldsymbol{\mathcal{T}} \in \mathfrak{R}^{C_{\mathcal{D}}}$

$$\mathbb{D}\boldsymbol{\mathcal{T}} \cdot \boldsymbol{\mathcal{T}} \geq 0. \tag{A.1}$$

To prove this results, let us write the $c$th entry of vector $\mathbb{D}\boldsymbol{\mathcal{T}}$

$$\begin{aligned}(\mathbb{D}\boldsymbol{\mathcal{T}})_c &= \sum_{p\in\mathcal{P}(c)} Q_{pc} \\ &= \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{F}(p,c)} A^f_{pc}q^f_{pc}.\end{aligned}$$

Employing the above expression, the left-hand side of (A.1) reads

$$\mathbb{D}\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}} = \sum_{c=1}^{\mathsf{C}_\mathcal{D}}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{F}(p,c)} A^f_{pc}q^f_{pc}T_c.$$

Interchanging the order of summation lead to

$$\begin{aligned}\mathbb{D}\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}} &= \sum_{p=1}^{\mathsf{P}_\mathcal{D}}\sum_{c\in\mathcal{C}(p)}\sum_{f\in\mathcal{F}(p,c)} A^f_{pc}q^f_{pc}T_c \\ &= \sum_{p=1}^{\mathsf{P}_\mathcal{D}} I_p.\end{aligned}$$

Here, $\mathsf{P}_\mathcal{D}$ is the total number of nodes of the computational grid and $I_p = \sum_{c\in\mathcal{C}(p)}\sum_{f\in\mathcal{F}(p,c)} A^f_{pc}q^f_{pc}T_c$ has been already defined by Eq. (3.19). Due to the fundamental inequality satisfied by the discrete sub-face normal flux approximation, refer to Section 3.3, $I_p$ is always positive, which ends the proof.

## A.2 $L^2$-stability of the semi-discrete scheme

In this paragraph, we prove the stability of our semi-discrete scheme in the absence of source term ($r=0$) with respect to the discrete $L^2$ weighted norm defined by

$$\|\boldsymbol{\mathcal{T}}\|^2_{w2} = \sum_{c=1}^{\mathsf{C}_\mathcal{D}} m_c C_{cv} T_c^2, \tag{A.2}$$

where $\mathsf{C}_\mathcal{D}$ is the total number of cells of the computational domain $\mathcal{D}$. In the absence of the source term, the semi-discrete scheme reads

$$\mathbb{M}\mathbb{C}_v\frac{\mathrm{d}\boldsymbol{\mathcal{T}}}{\mathrm{d}t} + \mathbb{D}\boldsymbol{\mathcal{T}} = \mathbf{0},$$

Dot-multiplying the above equation by $\boldsymbol{\mathcal{T}}\in\mathfrak{R}^{\mathsf{C}_\mathcal{D}}$ yields

$$\mathbb{M}\mathbb{C}_v\frac{\mathrm{d}\boldsymbol{\mathcal{T}}}{\mathrm{d}t}\cdot\boldsymbol{\mathcal{T}} + \mathbb{D}\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}} = 0.$$

Assuming that the mass density and the heat capacity do not depend on time, the above equation turns into

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\mathbb{M}\mathbb{C}_v\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}}\right)=-\mathbb{D}\boldsymbol{\mathcal{T}}\cdot\boldsymbol{\mathcal{T}}.$$

Recalling that the global diffusion matrix, $\mathbb{D}$, is positive semi-definite and employing the definition of the discrete $L^2$ norm (A.2) leads to the inequality

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\|\boldsymbol{\mathcal{T}}\|_{w2}^2\right)\leq 0. \tag{A.3}$$

Here, we have ignored the contributions of the boundary terms assuming for instance periodic or homogeneous Neumann boundary conditions. This inequality shows that the $L^2$ norm of the semi-discrete solution remains bounded by the $L^2$ norm of the initial data. This implies the $L^2$-stability of our semi-discrete finite volume scheme.

## A.3 Boundary conditions

In this paragraph, we present a generic methodology to implement the boundary conditions, which is crucial when dealing with real-word applications. It is worth mentioning that the boundary terms discretization is derived in a consistent manner with the scheme construction. To take into account the boundary terms, let us write the linear system linking the sub-face temperatures with the cell temperature under the form

$$\mathbb{N}\bar{\boldsymbol{T}}=\mathbb{S}\boldsymbol{T}+\boldsymbol{B}, \tag{A.4}$$

where the extra term $\boldsymbol{B}$ is the vector containing the boundary conditions contribution, which shall be defined in the next paragraphs.

Let us consider a sub-face $f$ located on the boundary of the domain, in the next paragraphs, we describe the modifications to bring to the matrices and boundary vector, depending on the boundary conditions types under consideration.

### A.3.1 Dirichlet boundary condition

On the boundary sub-face $f\equiv(c,i)$, the temperature $\bar{T}^*$ is imposed, we have $\bar{T}_c^i=\bar{T}^f=\bar{T}^*$. We multiply this equation by $A_c^i$, thus $A_c^i\bar{T}^f=A_c^i\bar{T}^*$. Let us write this equation under the system form (A.4). The diagonal term of the $f$th line of the system writes

$$\mathbb{N}_{ff}=A_c^i.$$

The corresponding extra-diagonal term is given by

$$\mathbb{N}_{fg}=0,\quad \forall g\neq f.$$

Regarding the matrix $\mathbb{S}$, we obtain

$$\mathbb{S}_{fg}=0,\quad \forall g.$$

Finally, the $f$th component of the vector $\boldsymbol{B}$ reads

$$\boldsymbol{B}_f=A_c^i\bar{T}^*.$$

### A.3.2 Neumann boundary condition

On the boundary sub-face $f\equiv(c,i)$ the normal flux $q^*$ is prescribed, hence the continuity condition rewrites

$$q_c^i = q^*. \tag{A.5}$$

Multiplying this equation by $A_c^i$ and replacing $q_c^i$ by its expression (3.26) yields

$$-\alpha_c A_c^i \sum_{k=1}^{3} \boldsymbol{K}_{ik}^c A_c^k (\bar{T}_c^k - T_c) = A_c^i q^*. \tag{A.6}$$

The diagonal term of the $f$th line of matrix $\mathbb{N}$ reads

$$\mathbb{N}_{ff} = \alpha_c A_c^i \boldsymbol{K}_{ii}^c A_c^i.$$

There are two non-zero extra-diagonal terms that come from the contribution of the sub-cell $c$. If we note $g\equiv(c,k)$, for $k\neq i$, these two terms write under the form

$$\mathbb{N}_{fg} = \alpha_c A_c^i \boldsymbol{K}_{ik}^c A_c^k.$$

The matrix $\mathbb{S}$ has only one non-zero term its $f$th line

$$\mathbb{S}_{fc} = \alpha_c \sum_{k=1}^{3} A_c^i \boldsymbol{K}_{ik}^c A_c^k.$$

Finally, the $f$th component of vector $\boldsymbol{B}$ is $\boldsymbol{B}_f = -A_c^i q^*$.

### A.3.3 Robin boundary condition

On the boundary sub-face $f\equiv(c,i)$, the condition $\alpha\bar{T}_c^i + \beta q_c^i = q_R^*$ is prescribed. Let us multiply this equation by $A_c^i$ and replace $q_c^i$ by its expression (3.26) to obtain

$$\alpha A_c^i \bar{T}_c^i - \beta\alpha_c A_c^i \sum_{k=1}^{3} \boldsymbol{K}_{ik}^c A_c^k (\bar{T}_c^k - T_c) = A_c^i q_R^*. \tag{A.7}$$

The diagonal term of matrix $\mathbb{N}$ reads

$$\mathbb{N}_{ff} = \beta\alpha_c A_c^i \boldsymbol{K}_{ii}^c A_c^i - \alpha A_c^i.$$

This matrix has once again two non-zero extra-diagonal terms coming from the contribution of the sub-cell $c$. Denoting $g\equiv(c,k)$, for $k\neq i$, these two non-zero terms write

$$\mathbb{N}_{fg} = \beta\alpha_c A_c^i \boldsymbol{K}_{ik}^c A_c^k.$$

The non-zero term of the $f$th line of matrix $\mathbb{S}$ is given by

$$\mathbb{S}_{fc} = \beta\alpha_c \sum_{k=1}^{3} A_c^i \boldsymbol{K}_{ik}^c A_c^k.$$

Finally, the $f$th component of vector $\boldsymbol{B}$ is $\boldsymbol{B}_f = -A_c^i q_R^*$.

Let us remark that the Dirichlet boundary condition is recovered for $\alpha=1$, $\beta=0$ and $q_R^* = T^*$ whereas, the Neumann boundary condition corresponds to the case $\alpha=0$, $\beta=1$ and $q_R^* = q^*$.

### A.3.4 Contribution to the global diffusion matrix

We achieve the discretization of the boundary conditions by listing the modifications that we have to take into account in the assembling of the global diffusion matrix. Solving the local system (A.4), which relates the sub-face temperatures and the cell temperatures, yields the following expression of the sub-face temperature vector

$$\bar{\boldsymbol{T}} = \mathbb{N}^{-1}\mathbb{S}\boldsymbol{T} + \mathbb{N}^{-1}\boldsymbol{B}, \tag{A.8}$$

where the modifications inherent to matrices $\mathbb{N}$, $\mathbb{S}$ and vector $\boldsymbol{B}$ have been detailed in the previous paragraphs. The above expression of the sub-face temperature vector, $\bar{\boldsymbol{T}}$, turns the contribution of the sub-cell $\omega_{pc}$ to the diffusion flux, $Q_{pc}$, into

$$Q_{pc} = -\sum_{d\in\mathcal{C}(p)} \mathbb{G}^p_{cd}(T_d - T_c) - \left(\widetilde{\mathbb{S}}^t\mathbb{N}^{-1}\boldsymbol{B}\right)_c, \tag{A.9}$$

where the effective conductivity tensor, $\mathbb{G}^p$, is defined by (3.33). Finally, the global linear system corresponding to our finite volume scheme becomes

$$\mathbb{M}\mathbb{C}_v\frac{\mathrm{d}\boldsymbol{\mathcal{T}}}{\mathrm{d}t} + \mathbb{D}\boldsymbol{\mathcal{T}} = \mathbb{M}\boldsymbol{\mathcal{R}} + \boldsymbol{\Sigma}, \tag{A.10}$$

where $\boldsymbol{\Sigma}$ is the vector containing the boundary condition contributions, whose the $c$th entry is given by $\boldsymbol{\Sigma}_c = \left(\widetilde{\mathbb{S}}^t\mathbb{N}^{-1}\boldsymbol{B}\right)_c$. The definition of the other matrices and vectors of the above system remain unchanged.

### A.3.5 Comment about the size of the local node-based linear systems satisfied by the sub-face temperatures

Let us recall that the pre-processing step for eliminating sub-face temperatures requires solving a set of local sparse linear systems at nodes, refer to Section 3.5. The size of these systems, which remains small compared to the size of global linear system, is equal to the number of faces impinging at nodes. For instance, in a Cartesian structured grid the size of these systems is constant and equal to 12x12 since at a given node the number of impinging faces is equal to 12. In an unstructured tetrahedral grid the size of these systems may vary a lot since the number of faces impinging at a node is a function of the grid connectivity. To illustrate this point, we have counted the minimum and the maximum size of these local node-based linear systems for a sequence of four tetrahedral grids which have been used in the numerical tests to pave a truncated sphere, refer to Fig. 10(f) in Section 6. We observe in Table 7 that the size of these systems remains small compared to the size of the global system which is nothing but the number of grid cells. Indeed the size of the local node-based linear systems ranges from 5 to 102 and the averaged size is below 30.

Table 7: Statistics about the size of the local node-based systems for a sequence of refined tetrahedral grids.

| Number of grid cells | Number of grid nodes | Minimum size | Maximum size | Averaged size |
|---|---|---|---|---|
| 902 | 189 | 7 | 66 | 21 |
| 6623 | 1219 | 7 | 96 | 27 |
| 13549 | 2447 | 7 | 102 | 28 |
| 37648 | 6543 | 5 | 90 | 30 |

# B Modifications to take into account pyramid cells

Pyramid cells are required to construct a conformal partition of a computational domain made of tetrahedral and hexahedral cells. Indeed, the pyramid cells allow to make the transition between the tetrahedral zones and the hexahedral zones. In this case, we have to slightly modify our finite volume scheme to take into account the fact that pyramids are cells for which the number of faces incident to one vertex is strictly greater than 3. We describe the needed modifications by considering a generic pyramid $\omega_c$ and we denote by $p$ the vertex characterized by $\mathsf{F}_{pc}=4$, where $\mathsf{F}_{pc}$ denotes the number of faces of cell $c$ impinging at point $p$, refer to Fig. 12. Knowing that $\mathsf{F}_{pc}=4$ faces are incident to the vertex $p$, the decomposition of a vector in terms of its normal components within sub-cell $\omega_{pc}$, refer to Section 3.1, is not possible. Indeed, the number of equations, *i.e.*, $\mathsf{F}_{pc}=4$, being greater than the number of unknowns, *i.e.*, the 3 Cartesian components of the vector under consideration, we end up with an overdetermined system.

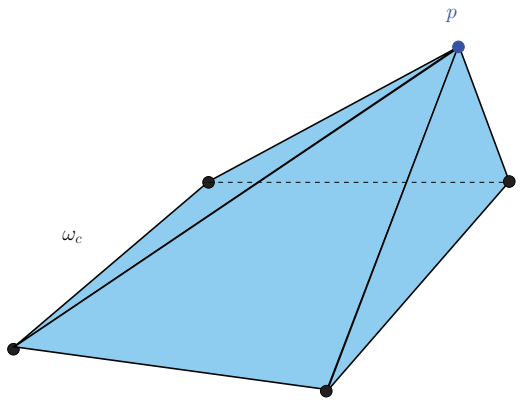



Figure 12: Sketch of a pyramid cell.

To overcome this difficulty, we subdivide the sub-cell $\omega_{pc}$ into the $\mathsf{F}_{pc}=4$ fictive sub-cells $\omega_{pcf}$ defined by

$$\omega_{pcf}=\bigcup_{e\in\mathcal{E}(p,f)}\mathcal{I}^{pfe},\quad \text{for } f\in\mathcal{F}(p,c).$$

Here, $\mathcal{E}(p,f)$ is the set of edges of face $f$ impinging at point $p$. Namely, being given a

face $f$ incident to the vertex $p$, the sub-cell $\omega_{pcf}$ is constructed by gathering the two iota tetrahedra attached to the two edges of face $f$ incident to point $p$. We observe that there is one fictive sub-cell, $\omega_{pcf}$, per face impinging at vertex $p$. Each fictive sub-cell $\omega_{pcf}$ has 3 faces impinging at node $p$: the outer sub-face $\partial\omega_{pc}^{f}$ and two inner sub-faces which result from the subdivision. Bearing this in mind, we can employ (3.26) to write the flux approximation within each fictive sub-cell $\omega_{pcf}$. Having added the supplementary fictive sub-cells, the number of sub-cells surrounding point $p$, which was equal to $\mathsf{C}_p$, becomes equal to $\mathsf{C}_p^{\triangle}=\mathsf{C}_p+\mathsf{F}_{pc}-1$. Here, without loss of generality, we suppose that there is only one pyramid in the set of cells surrounding vertex $p$. Regarding the number of faces incident to vertex $p$, it was equal to $\mathsf{F}_p$ and becomes equal to $\mathsf{F}_p^{\triangle}=\mathsf{F}_p+\mathsf{F}_{pc}$. Therefore, at the vertex $p$, the vector of sub-face temperatures, $\bar{\boldsymbol{T}}^{\triangle}$, is of size $\mathsf{F}^{\triangle}$ and the vector of cell-centered temperatures, $\boldsymbol{T}^{\triangle}$, is of size $\mathsf{C}_p^{\triangle}$. Utilizing the flux approximation (3.26) and enforcing the normal flux continuity across the cell interfaces surrounding vertex $p$ in the same manner than in Section 3.5 leads to the linear system satisfied by the sub-face temperatures

$$\mathbb{N}^{\triangle}\bar{\boldsymbol{T}}^{\triangle}=\mathbb{S}^{\triangle}\boldsymbol{T}^{\triangle}.$$

Here, $\mathbb{N}^{\triangle}$ and $\mathbb{S}^{\triangle}$ are respectively matrices of size $\mathsf{F}_p^{\triangle}\times\mathsf{F}_p^{\triangle}$ and $\mathsf{F}_p^{\triangle}\times\mathsf{C}_p^{\triangle}$ which are constructed in the same way than in Section 3.5. The matrix $\mathbb{N}^{\triangle}$ is invertible, refer to Section 3.6, and the solution of the above linear system writes

$$\bar{\boldsymbol{T}}^{\triangle}=\left(\mathbb{N}^{\triangle}\right)^{-1}\mathbb{S}^{\triangle}\boldsymbol{T}^{\triangle}.$$

This formula allows to express the sub-face temperatures $p$ in terms of the cell-centered temperatures surrounding vertex $p$. Finally, using the same procedure than in Section 3.7, the contribution of cell $c$ to the diffusion flux at vertex $p$ writes

$$Q_{pc}=-\sum_{d\in\mathcal{C}^{\triangle}(p)}\mathbb{G}_{cd}^{p,\triangle}(T_d^{\triangle}-T_c^{\triangle}), \tag{B.1}$$

where $\mathcal{C}^{\triangle}(p)$ is the set of cells surrounding vertex $p$ including the fictive sub-cells. The $\mathsf{C}_p^{\triangle}\times\mathsf{C}_p^{\triangle}$ matrix $\mathbb{G}^{p,\triangle}$ is given by $\mathbb{G}^{p,\triangle}=\left(\widetilde{\mathbb{S}}^{\triangle}\right)^{t}\left(\mathbb{N}^{\triangle}\right)^{-1}\mathbb{S}^{\triangle}$, refer to Section 3.7 for the definition of $\widetilde{\mathbb{S}}$. We point out that the cell index, $d$, employed in (B.1), can refer to a fictive sub-cell. More precisely, $Q_{pc}$ contains contributions coming from temperatures attached to the fictive sub-cells. These supplementary degrees of freedom are eliminated equating them to the cell temperature $T_c$. This amounts to express the vector of the cell-centered temperatures including the temperatures of the fictive sub-cells, $\boldsymbol{T}^{\triangle}\in\mathfrak{R}^{\mathsf{C}_p^{\triangle}}$, in terms of the initial vector of the cell-centered temperatures $\boldsymbol{T}\in\mathfrak{R}^{\mathsf{C}_p}$ as follows

$$\boldsymbol{T}^{\triangle}=\mathbb{P}\boldsymbol{T}. \tag{B.2}$$

Here, $\mathbb{P}$ is a rectangular matrix of size $\mathsf{C}_p^{\triangle}\times\mathsf{C}_p$. Let $i$ (resp. $j$) be the generic index of a cell in the local numbering of the cells belonging to $\mathcal{C}^{\triangle}(p)$ (resp. $\mathcal{C}(p)$), then according to

(B.2), temperature $T_i^{\triangle}$ writes

$$T_i^{\triangle} = \sum_{j=1}^{\mathsf{C}_p} \mathbb{P}_{ij} T_j.$$

For $i = 1, \cdots, \mathsf{C}_p^{\triangle}$ and $j = 1, \cdots, \mathsf{C}_p$, the generic entry of $\mathbb{P}$ writes

$$\mathbb{P}_{ij} = \begin{cases} 1, & \text{if } i \text{ corresponds to a fictive sub-cell of } c \text{ and } j \text{ corresponds to cell } c, \\ 1, & \text{if } i \text{ corresponds to cell } c \text{ and } j \text{ corresponds to cell } c, \\ 0, & \text{elsewhere.} \end{cases}$$

Finally, substituting (B.2) into (B.1) leads to the expression of $Q_{pc}$ in terms of cell-centered temperatures

$$Q_{pc} = -\sum_{d \in \mathcal{C}(p)} \mathbb{G}_{cd}^{p} (T_d - T_c), \tag{B.3}$$

where $\mathbb{G}_{cd}^{p} = \mathbb{P}^t \mathbb{G}^{p,\triangle} \mathbb{P}$. It is worth pointing out that the definition of the global diffusion matrix remains unchanged.

We have described the above modification in the particular case of a pyramid but there is nothing to prevent us from applying it to general polyhedral cells.

## References


[1] PlaFrim Web page. Available at `https://plafrim.bordeaux.inria.fr/doku.php?id=start`.

[2] I. Aavatsmark, T. Barkve, O. Boe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anistropic media. Part I: derivation of the methods. SIAM J. Sci. Comput., 19:1700–1716, 1998.

[3] I. Aavatsmark, T. Barkve, O. Boe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anistropic media. Part II: discussion and numerical results. SIAM J. Sci. Comput., 19:1717–1736, 1998.

[4] I. Aavatsmark, T. Barkve, and T. Mannseth. Control volume discretization methods for 3D quadrilateral grids in inhomogeneous, anisotropic reservoirs. SPE J., pages 146–154, 1998.

[5] I. Aavatsmark, G. T. Eigestad, B.-O. Heimsund, B. T. Mallison, J. M. Nordbotten, and E. Øian. A new finite volume approach to efficient discretization on challenging grids. In Proceedings of the SPE International Reservoir Simulation Symposium, number SPE 106435, Houston, USA, 2007.

[6] I. Aavvatsmark, G. T. Eigestad, R. A. Klausen, M. F. Wheeler, and I. Yotov. Convergence of a symmetric MPFA method on quadrilateral grids. Technical Report TR-MATH 05-14, University of Pittsburgh, 2005.

[7] L. Agelas and R. Masson. Convergence of the finite volume mpfa o scheme for heterogeneous anisotropic diffusion problems on general meshes. Comptes Rendus Mathematique, 346:1007–1012, 2008.

[8] G. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. AFIPS Conference Proceedings, (30):483–485, 1967.

[9] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2012. Available at `http://www.mcs.anl.gov/petsc`.
[10] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.
[11] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997.
[12] D. Bianchi, F. Nasuti, and E. Martelli. Navier-Stokes Simulations of Hypersonic Flows with Coupled Graphite Ablation. Journal of Spacecraft and Rockets, 47(4), 2010.
[13] J. Breil and P.-H. Maire. A cell-centered diffusion scheme on two-dimensional unstructured meshes. J. Comp. Phys., 224(2):785–823, 2007.
[14] F. Brezzi, K. Lipnikov, M. Shashkov, and V. Simoncini. A new discretization methodology fro diffusion problems on generalized polyhedral meshes. Comput. Methods Appl. Mech. Engrg., 196:3682–3692, 2007.
[15] D. E. Burton. Multidimensional Discretization of Conservation Laws for Unstructured Polyhedral Grids. Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
[16] K. Domelovo and P. Omnes. A finite volume for the laplace equation on almost arbitrary two-dimensional grids. Mathematical Modelling and Numerical Analysis, 39(6):1203–1249, 2005.
[17] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009.
[18] D. Gunasekera, P. Childs, J. Herring, and J. Cox. A multi-point flux discretization scheme for general polyhedral grids. In Proceedings of the SPE International Oil and Gas Conference and Exhibition in China, number SPE 48855, Beijing, China, 1998.
[19] F. Hermeline. A finite volume method for the approximation of diffusion operators on distorded meshes. J. Comp. Phys., 160:481–499, 2000.
[20] F. Hermeline. Approximation of 2-d and 3-d diffusion operators with variable full tensor coefficients on arbitrary meshes. Comput. Methods Appl. Mech. Engrg., 196:2497–2526, 2007.
[21] F. Hermeline. A finite volume method for approximating 3D diffusion operators on general meshes. J. Comp. Phys., 228:5763–5786, 2009.
[22] F. Hermeline. Un point sur les méthodes DDFV, 2010. Advanced methods for the diffusion equation on general meshes; Université Pierre et Marie Curie, Paris France, July 2010; available at `http://www.ann.jussieu.fr/~despres/WEB/Talks/hermeline.pdf`.
[23] J. Hyman, J.E. Morel, M. Shashkov, and S. Steinberg. Mimetic finite difference methods for diffusion equations. Computational Geosciences, 6:333–352, 2002.
[24] J. Hyman, M. Shashkov, and S. Steinberg. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. J. Comp. Phys., 132:130–148, 1997.
[25] K. Lipnikov, J. E. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-othogonal non-conformal meshes. J. Comp. Phys., 199:589–597, 2004.
[26] K. Lipnikov, M. Shashkov, and D. Svyatskiy. The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes. J. Comp. Phys., 211:473–491, 2006.

[27] K. Lipnikov, M. Shashkov, and I. Yotov. Local flux mimetic finite difference methods. Numerische Mathematik, 112(1):115–152, 2009.
[28] P.-H. Maire and J. Breil. A high-order finite volume cell-centered scheme for anisotropic diffusion on two-dimensional unstructured grids. J. Comp. Phys., 224(2):785–823, 2011.
[29] J. E. Morel, R. M. Roberts, and M. Shashkov. A local support-operators diffusion discretization scheme for quadrilateral r-z meshes. J. Comp. Phys., 144:17–51, 1998.
[30] K. Nakajima, H. Nakamura, and T. Tanahashi. Parallel iterative solvers with localized ilu preconditioning. In Bob Hertzberger and Peter Sloot, editors, High-Performance Computing and Networking, volume 1225 of Lecture Notes in Computer Science, pages 342–350. Springer Berlin Heidelberg, 1997.
[31] J. T. Oden, I. Babuska, and C. E. Baumann. A Discontinuous hp Finite Element Method for Diffusion problems. J. Comp. Phys., 146:491–519, 1998.
[32] M. Pal and M. G. Edwards. Quasi monotonic continuous darcy-flux approximation for general 3-d gids on any element type. In Proceedings of the SPE International Reservoir Simulation Symposium, number SPE 106486, Houston, USA, 2007.
[33] F. Pellegrini. Scotch Web page, 2012. Available at `https://gforge.inria.fr/projects/scotch/`.
[34] C. Le Potier. Schémas volumes finis pour des opérateurs de diffusion fortement anisotropes sur des maillages non structurés. Comptes Rendus Mathematique, 340:921–926, 2005.
[35] R. Rieben and D. White. Verification of high-order mixed finite element solution of transient magnetic diffusion problems. IEEE Transactions on Magnetics, 42(1):25–39, 2006.
[36] M. Shashkov and S. Steinberg. Support-Operator Finite-Difference Algorithms for General Elliptic Problems. J. Comp. Phys., 118:131–151, 1995.
[37] M. Shashkov and S. Steinberg. Solving Diffusion Equations with Rough Coefficients in Rough Grids. J. Comp. Phys., 129:383–405, 1996.
[38] J.-M. Thomas and D. Trujillo. Mixed finite volume methods. International Journal for Numerical Methods in Engineering, 46:1351–1366, 1999.
[39] H. A. van der Vorst. Bi-CGStab: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. SIAM J. Sci. and Stat. Comput., 13(2):631–644, 1992.
[40] M. Vohralik. Equivalence between lowest-order mixed finite element and multi-point finite volume methods on simplicial meshes. Mathematical Modelling and Numerical Analysis, 40(2):367–391, 2006.
[41] L. T. Yang and R. P. Brent. The improved BiCGStab method for large and sparse unsymmetric linear systems on parallel distributed memory architectures. In Algorithms and Architectures for Parallel Processing, 2002. Proceedings. Fifth International Conference on, pages 324 –328, oct. 2002.